

[illegible]

```
CCCCCCCC  RRRRRRRR  EEEEEEEEE  TTTTTTTTT  RRRRRRRR  AAAAAA  NN  NN  SSSSSSSS
CCCCCCCC  RRRRRRRR  EEEEEEEEE  TTTTTTTTT  RRRRRRRR  AAAAAA  NN  NN  SSSSSSSS
CC         RR      RR  EE         TT         RR      AA      AA  NN  NN  SS
CC         RR      RR  EE         TT         RR      AA      AA  NN  NN  SS
CC         RR      RR  EE         TT         RR      AA      AA  NNNN  NN  SS
CC         RRRRRRRR  EEEEEEEE  TT         RRRRRRRR  AA      AA  NN  NN  SSSSSS
CC         RRRRRRRR  EEEEEEEE  TT         RRRRRRRR  AA      AA  NN  NN  SSSSSS
CC         RR      RR  EE         TT         RR      AAAAAAAAAA  NN  NNNN  SS
CC         RR      RR  EE         TT         RR      AAAAAAAAAA  NN  NNNN  SS
CC         RR      RR  EE         TT         RR      AA      AA  NN  NN  SS
CC         RR      RR  EE         TT         RR      AA      AA  NN  NN  SS
CCCCCCCC  RR      RR  EEEEEEEEE  TT         RR      AA      AA  NN  NN  SSSSSSSS
CCCCCCCC  RR      RR  EEEEEEEEE  TT         RR      AA      AA  NN  NN  SSSSSSSS

LL         IIIIII  SSSSSSSS
LL         IIIIII  SSSSSSSS
LL         II      SS
LL         II      SS
LL         II      SS
LL         II      SS
LL         II      SSSSSS
LL         II      SSSSSS
LL         II      SS
LL         II      SS
LL         II      SS
LL         II      SS
LLLLLLLLLL  IIIIII  SSSSSSSS
LLLLLLLLLL  IIIIII  SSSSSSSS
```

.....



```
0001 0 MODULE CRE_TRANS(IDENT='V04-000') = ! File: SORRMSIO.B32 Edit: PDG3006
0002 1 BEGIN
0003 1
0004 1 *****
0005 1 *
0006 1 * COPYRIGHT (c) 1978, 1980, 1982, 1984 BY *
0007 1 * DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS. *
0008 1 * ALL RIGHTS RESERVED. *
0009 1 *
0010 1 * THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED *
0011 1 * ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE *
0012 1 * INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER *
0013 1 * COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY *
0014 1 * OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY *
0015 1 * TRANSFERRED. *
0016 1 *
0017 1 * THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE *
0018 1 * AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT *
0019 1 * CORPORATION. *
0020 1 *
0021 1 * DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS *
0022 1 * SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL. *
0023 1 *
0024 1 *
0025 1 *****
0026 1
0027 1 ++
0028 1
0029 1 TITLE: CRE_TRANS CDD COBOL Language Extractor
0030 1
0031 1 FACILITY: Common Data Dictionary
0032 1
0033 1 ABSTRACT:
0034 1
0035 1 This module contains the routines used to translate CDD definitions.
0036 1 It can translate the following types of definitions:
0037 1
0038 1 Records
0039 1
0040 1 ENVIRONMENT:
0041 1
0042 1 All routines run in user access mode.
0043 1
0044 1 These routines are NOT AST reentrant.
0045 1
0046 1 AUTHOR: Jeff East, 30-Jan-81
0047 1
0048 1 MODIFIED BY:
0049 1
0050 1 T03-002 Removed copious amounts of unneeded code. PDG 3-Jan-1983
0051 1 T03-003 Use UTIL$FIND SYMBOL to reference CDD symbols. PDG 2-Mar-1983
0052 1 T03-004 Add DSC$K DTYPE B(U), which was overlooked. PDG 1-Jul-1983
0053 1 T03-005 Use LIB$FIND IMAGE SYMBOL, instead. PDG 9-Mar-1984
0054 1 T03-006 Change attributes lists from OWN to LOCAL. PDG 10-Apr-1984
0055 1 --
```



```
57 0056 1 LIBRARY 'SYSS$LIBRARY:STARLET';
58 0057 1 REQUIRE 'SRC$:COM';
59 0127 1 REQUIRE 'SRC$:CDDMAC'; ! REQUIRE 'CDD$LIBRARY:CDDMAC';
60 1323 1 REQUIRE 'SRC$:CDDTYPE'; ! REQUIRE 'CDD$LIBRARY:CDDTYPE';
61 1477 1 REQUIRE 'SRC$:REQSYM'; ! REQUIRE 'CDD$PROTOCOL:REQSYM';
62 1533 1 REQUIRE 'SRC$:RECSYM'; ! REQUIRE 'CDD$PROTOCOL:RECSYM';
63 1664 1
64 1665 1 LITERAL
65 1666 1 REG_PRM = 10;
66 1667 1 LINKAGE
67 1668 1 CAL_CTXREG_LB = CALL: GLOBAL(CTX=COM_REG_CTX,PRM=REG_PRM);
68 1669 1
69 1670 1 FORWARD ROUTINE
70 1671 1 CRE_DIAG: CAL_CTXREG,
71 1672 1 FREE_DYN,
72 1673 1 CRE_FIELD: CAL_CTXREG_LB,
73 1674 1 CRE_OCCURS_CLAUSE: CAL_CTXREG,
74 1675 1 CRE_OVERLAY_FIELDS: CAL_CTXREG_LB,
75 1676 1 CRE_TRANS_RECORD: CAL_CTXREG_LB NOVALUE,
76 1677 1 CRE_HISTORY: CAL_CTXREG NOVALUE,
77 1678 1 CRE_EXTRACT_RECORD: CAL_CTXREG,
78 1679 1 CRE_CLEAN_UP: CAL_CTXREG NOVALUE,
79 1680 1 CDD$CREATE_HISTORY,
80 1681 1 CDD$FIND_NODE,
81 1682 1 CDD$GET_ATT,
82 1683 1 CDD$GET_ENTITY_CELL,
83 1684 1 CDD$GET_ENTITY_LIST_ATT,
84 1685 1 CDD$GET_NUM_ATT,
85 1686 1 CDD$GET_STRING_ATT,
86 1687 1 CDD$PLSE_LOCKS,
87 1688 1 CDD$SIGN_IN,
88 1689 1 CDD$SIGN_OUT,
89 1690 1 CDD_ATTNOTFND: NOVALUE;
90 1691 1
91 1692 1 SOR$SEND_ROUTINE_(CRE_CLEAN_UP);
92 1693 1
93 1694 1 EXTERNAL ROUTINE
94 1695 1 SOR$DEF_FIELD: CAL_CTXREG NOVALUE,
95 1696 1 SOR$ERROR,
96 1697 1 STR$FREE1_DX: NOVALUE ADDRESSING_MODE(GENERAL);
97 1698 1
98 1699 1 Error messages
99 1700 1
100 1701 1 SOR$_BADDTYPE !E Invalid or unsupported CDD datatype
101 1702 1 SOR$_BADLENOFF !E Length and offset must be multiples of 8 bits
102 1703 1 SOR$_BADOCCURS !F Invalid OCCURS clause
103 1704 1 SOR$_BADOVLAY !E Invalid overlay structure
104 1705 1 SOR$_BADPROTCL !F Node is an invalid CDD object
105 1706 1 SOR$_CDDERROR !F CDD error at node !AS
106 1707 1 SOR$_ILLBASE !E Nondecimal base is invalid
107 1708 1 SOR$_ILLLITERL !F Record containing symbolic literals is unsupported
108 1709 1 SOR$_ILLSCALE !E Nonzero scale invalid for floating point data-item
109 1710 1 SOR$_INCDIGITS !E Number of digits is not consistent with the type or length of item
110 1711 1 SOR$_MISLENOFF !W Length and offset required
111 1712 1 SOR$_MULTIDIM !E Invalid multidimensional OCCURS
112 1713 1 SOR$_NOTRECORD !F Node !AS is a !AS, not a record definition
113 1714 1 SOR$_SIGNCOMPQ !W Absolute Date and Time datatype represented in one second units
```



CRE\_TRANS  
V04=000

K 1  
16-Sep-1984 01:08:56  
14-Sep-1984 13:10:31

VAX-11 Bliss-32 V4.0-742  
[SORT32.SRC]CRETRANS.B32;1

Page 3  
(2)

User Literal Definitions

```
: 114      1715 1 !      SOR$_UNSUPLEVL  !F Unsupported core level for record !AS
: 115      1716 1
: 116      1717 1
: 117      1718 1 MACRO
: 118      1719 1      CDD$_ATTNOTFND = .CDD_ ATTNOTFND %;
: 119      1720 1 OWN   CDD_--ATTNOTFND: INITIAC(0);
: 120      1721 1
: 121      1722 1 MACRO
: 122      1723 1      COM_CDD_PATH = 0+%FIELDEXPAND(COM_CDD,0), 0, 32, 0 %;
: 123      1724 1      COM_CDD_CTX = 1+%FIELDEXPAND(COM_CDD,0), 0, 32, 0 %;
: 124      1725 1 ASSERT_(COM_K_CDD GEQ 2)
```



```
: 126      1726 1 MACRO
: 127      1727 1
: 128      1728 1 MACRO      $DSC = BLOCK[8,BYTE] %;
: 129      M 1729 1
: 130      M 1730 1          CRE_DECLARE_HANDLER[] =
: 131      1731 1              ENABLE FREE_DYN(%REMAINING);
: 132      1732 1 MACRO      $INIT_DSC(%REMAINING) %;
: 133      M 1733 1
: 134      M 1734 1          $INIT_DSC[DSC_NAM] =
: 135      M 1735 1              BEGIN
: 136      M 1736 1              DSC_NAM[DSC$B_DTYPE] = DSC$K_DTYPE_T;
: 137      M 1737 1              DSC_NAM[DSC$B_CLASS] = DSC$K_CLASS_D;
: 138      M 1738 1              DSC_NAM[DSC$W_LENGTH] = 0;
: 139      1739 1              DSC_NAM[DSC$A_POINTER] = 0;
:                          END %;
```



## User Literal Definitions

```
141 1740 1 MACRO
142 1741 1
143 1742 1 !+
144 1743 1 $GET_ATTS_LIST (entry [, entry] ...)
145 1744 1
146 1745 1 Where:
147 1746 1
148 1747 1 entry = (name, attribute, type)
149 1748 1
150 1749 1 name = the name of the variable that is to receive the attribute
151 1750 1 value. The variable will be declared and initialized by
152 1751 1 this macro.
153 1752 1 attribute = the attribute name.
154 1753 1 entry = the type of attribute value:
155 1754 1
156 1755 1 ENTITY
157 1756 1 ENTITY_LIST
158 1757 1 NULL
159 1758 1 NUMERIC
160 1759 1 STRING
161 1760 1 STRING_LIST
162 1761 1
163 1762 1 This macro defines a macro called $GET_ATTS_INIT, which must be
164 1763 1 executed before the attribute list is used. The actual list is
165 1764 1 named ATTR_LIST.
166 1765 1 !-
167 1766 1
168 M 1767 1 $GET_ATTS_LIST [=
169 M 1768 1
170 M 1769 1 COMPILETIME
171 M 1770 1 ATTS_INDEX = 1;
172 M 1771 1
173 M 1772 1 $GET_ATTS_DEFINE (%REMAINING)
174 M 1773 1
175 M 1774 1 LOCAL
176 M 1775 1 ATTR_LIST: VECTOR[%LENGTH * 5 + 1, LONG]
177 M 1776 1 INITIAL (%LENGTH,
178 M 1777 1 $GET_ATTS_ENTRY (%REMAINING) );
179 M 1778 1
180 M 1779 1 %ASSIGN (ATTS_INDEX, 1)
181 M 1780 1 $GET_ATTS_ASSIGN (%REMAINING)
182 M 1781 1
183 M 1782 1 %ASSIGN (ATTS_INDEX, 1)
184 M 1783 1 MACRO
185 M 1784 1 %QUOTE %QUOTE $GET_ATTS_INIT =
186 M 1785 1 $GET_ATTS_INIT_ENTRY (%QUOTE %EXPAND %REMAINING)
187 M 1786 1 %QUOTE %;
188 1787 1 %,
189 1788 1
190 M 1789 1 $GET_ATTS_DEFINE[entry] =
191 M 1790 1 $GET_ATTS_DEF_ENTRY (%REMOVE (entry));
192 1791 1 %,
193 1792 1
194 M 1793 1 $GET_ATTS_DEF_ENTRY (name, attribute, type) =
195 M 1794 1 %IF NOT %IDENTICAL (type, %QUOTE NULL) %THEN
196 M 1795 1 %IF %IDENTICAL (type, %QUOTE STRING) %THEN
197 M 1796 1 LOCAL
```



```
198 M 1797 1 name: $DSC VOLATILE
199 M 1798 1 %ELSE
200 M 1799 1 LOCAL
201 M 1800 1 name: LONG
202 M 1801 1 %FI
203 M 1802 1 %FI
204 M 1803 1 %,
205 M 1804 1
206 M 1805 1 $GET_ATTS_ENTRY[entry] =
207 M 1806 1 $GET_ATTS_LIST_ENTRY (%REMOVE (entry))
208 M 1807 1 %,
209 M 1808 1
210 M 1809 1 $GET_ATTS_LIST_ENTRY (name, attribute, type) =
211 M 1810 1 attribute, %NAME ('CDD$k_', type), 0,
212 M 1811 1 %IF
213 M 1812 1 %IDENTICAL (type, %QUOTE NULL) OR %IDENTICAL (type, %QUOTE STRING)
214 M 1813 1 %THEN
215 M 1814 1 0
216 M 1815 1 %ELSE
217 M 1816 1 WORD (4, 0)
218 M 1817 1 %FI
219 M 1818 1 , 0
220 M 1819 1 %,
221 M 1820 1
222 M 1821 1 $GET_ATTS_ASSIGN[entry] =
223 M 1822 1 $GET_ATTS_ASG_ENTRY (%REMOVE (entry))
224 M 1823 1 %ASSIGN (ATTS_INDEX, ATTS_INDEX+5);
225 M 1824 1 %,
226 M 1825 1
227 M 1826 1 $GET_ATTS_ASG_ENTRY (name, attribute, type) =
228 M 1827 1 BIND
229 M 1828 1 %NAME (name, '_STATUS') = ATTR_LIST[%NUMBER(ATTS_INDEX)+2];
230 M 1829 1 %IF
231 M 1830 1 %IDENTICAL (type, %QUOTE ENTITY_LIST) OR
232 M 1831 1 %IDENTICAL (type, %QUOTE STRING_LIST)
233 M 1832 1 %THEN
234 M 1833 1 BIND
235 M 1834 1 %NAME (name, 'COUNT') =
236 M 1835 1 ATTR_LIST[%NUMBER(ATTS_INDEX)+3]+2: WORD;
237 M 1836 1 %FI
238 M 1837 1 %,
239 M 1838 1
240 M 1839 1 $GET_ATTS_INI_ENTRY[entry] =
241 M 1840 1 $GET_ATTS_INI_VAR (%REMOVE (entry))
242 M 1841 1 %ASSIGN (ATTS_INDEX, ATTS_INDEX+5);
243 M 1842 1 %,
244 M 1843 1
245 M 1844 1 $GET_ATTS_INI_VAR (name, attribute, type) =
246 M 1845 1 %IF NOT %IDENTICAL (type, %QUOTE NULL) %THEN
247 M 1846 1 %IF %IDENTICAL (type, %QUOTE STRING) %THEN
248 M 1847 1 $INIT_DSC (name);
249 M 1848 1 %FI
250 M 1849 1 ATTR_LIST[%NUMBER(ATTS_INDEX)+4] = name
251 M 1850 1 %FI
252 M 1851 1 %;
```



## User Literal Definitions

```
1852 1 ROUTINE CRE_DIAG(ERR, OPT): CAL_CTXREG =
1853 1
1854 1 ++
1855 1
1856 1 FUNCTIONAL DESCRIPTION:
1857 1
1858 1     This routine is called to issue a diagnostic on the COPY statement.
1859 1
1860 1 FORMAL PARAMETERS:
1861 1
1862 1     ERR      - error diagnostic to be issued.
1863 1     OPT      - zero or more optional parameters or additional messages
1864 1               to be included with the diagnostic.
1865 1
1866 1 IMPLICIT INPUTS:
1867 1
1868 1     NONE
1869 1
1870 1 IMPLICIT OUTPUTS:
1871 1
1872 1     NONE
1873 1
1874 1 RETURN VALUE:
1875 1
1876 1     Status value
1877 1
1878 1 SIDE EFFECTS:
1879 1
1880 1     NONE
1881 1
1882 1 --
1883 2 BEGIN
1884 2 EXTERNAL REGISTER
1885 2     CTX = COM_REG_CTX: REF CTX_BLOCK;
1886 2 LOCAL
1887 2     D: VECTOR[255+2];
1888 2 BUILTIN
1889 2     ACTUALCOUNT,
1890 2     CALLG;
1891 2
1892 2 D[0] = 0;
1893 2 D[1] = .ERR;
1894 2 D[2] = 0;
1895 2 IF .ERR NEQ SORS_CDDERROR THEN D[0] = 2;
1896 2
1897 2 D[.D[0]+1] = SORS_CDDERROR;
1898 2 D[.D[0]+2] = 1;
1899 2 D[.D[0]+3] = .CTX[COM_CDD_PATH];
1900 2 CH$MOVE((ACTUALCOUNT(5-1)*%UPVAL, OPT, D[.D[0]+4]));
1901 2 D[0] = .D[0] + ACTUALCOUNT() + 2;
1902 2 RETURN CALLG(D, SORS$error);
1903 2
1904 1 END;
```

```
.TITLE CRE_TRANS
.IDENT \V04-000\
```



```
.PSECT SOR$RO_CODE_____2,NOWRT, SHR, PIC,
00000000V 00000 _CRE_CLEAN_UP:
.LONG <CRE_CLEAN_UP-_CRE_CLEAN_UP> ;
.PSECT SOR$RW_PICDATA,NOEXE, PIC,2
00000000 00000 CDD__ATTNOTFND:
.LONG 0 ;
.EXTRN SOR$$DEF_FIELD, SOR$$ERROR
.EXTRN STR$FREET_DX
.PSECT SOR$RO_CODE,NOWRT, SHR, PIC,2
003C 00000 CRE_DIAG:
.WORD Save R2,R3,R4,R5 : 1852
MOVAB -1024(SP), SP : 1892
CLRL D : 1893
MOVL ERR, D+4 : 1894
CLRL D+8 : 1895
CML PL ERR, #1868348
BEQL 1$
MOVL #2, D : 1897
MOVL D, R0 : 1898
MOVL #1868348, D+4[R0] : 1899
MOVL D, R0 : 1900
MOVL #1, D+8[R0]
MOVL D, R0
MOVL 320(CTX), D+12[R0]
MOVZBL (AP), R1
DECL R1
MULL2 #4, R1
MOVL D, R0
PUSHAL D+16[R0]
MOVC3 R1, OPT, a(SP)+
MOVZBL (AP), R0 : 1901
ADDL2 D, R0
MOVL 2(R0), D
MOVAB D, SOR$$ERROR : 1902
RET : 1904
```

5E	FC00	CE	9E	00002	
		7E	D4	00007	
04	AE	04	AC	D0	00009
		08	AE	D4	0000E
001C823C	8F	04	AC	D1	00011
		03	13	00019	
6E		02	D0	0001B	
50		6E	D0	0001E	1\$:
04	AE40	001C823C	8F	D0	00021
50			6E	D0	0002A
08	AE40		01	D0	0002D
50			6E	D0	00032
0C	AE40	0140	CB	D0	00035
51			6C	9A	0003C
			51	D7	0003F
51			04	C4	00041
50			6E	D0	00044
		10	AE40	DF	00047
9E	08	AC	51	28	0004B
		50	6C	9A	00050
		50	6E	C0	00053
		02	A0	9E	00056
00000000G	00		6E	FA	0005A
			04	00061	

; Routine Size: 98 bytes, Routine Base: SOR\$RO\_CODE + 0000



```

: 308      1905 1 ROUTINE FREE_DYN(
: 309      1906 1     SIG: REF VECTOR,
: 310      1907 1     MCH: REF VECTOR,
: 311      1908 1     ENA: REF VECTOR) =
: 312      1909 2 BEGIN
: 313      1910 2 IF .SIG[1] EQL SSS_UNWIND
: 314      1911 2 THEN
: 315      1912 2     DECR I FROM .ENA[0] TO 1 DO
: 316      1913 2     IF ..ENA[I] NEQ 0 THEN STR$FREE1_DX(..ENA[I]);
: 317      1914 2 RETURN SSS_RESIGNAL;
: 318      1915 1 END;

```

0004 00000 FREE_DYN:						
				.WORD	Save R2	: 1905
				MOVL	SIG, R0	: 1910
00000920	50	04	AC D0 00002	CMPL	4(R0), #2336	
	8F	04	A0 D1 00006	BNEQ	3\$	
			1C 12 0000E	ADDL3	#1, @ENA, I	: 1912
52	0C	BC	01 C1 00010	BRB	2\$	: 1913
	50	0C	12 11 00015	MOVL	@ENA[I], R0	
			60 D5 0001C	TSTL	(R0)	
			09 13 0001E	BEQL	2\$	
			60 DD 00020	PUSHL	(R0)	
00000000G	00		01 FB 00022	CALLS	#1, STR\$FREE1_DX	
	EB		52 F5 00029	SOBGTR	I, 1\$	
	50	0918	8F 3C 0002C	MOVZWL	#2328, R0	: 1914
			04 00031	RET		: 1915

; Routine Size: 50 bytes, Routine Base: SOR\$R0\_CODE + 0062



## User Literal Definitions

```
320 1916 1 ROUTINE CRE_FIELD(FIELD_LOC, ROOT, FIRST_OFFSET, GROUP_OFFSET): CAL_CTXREG_LB =
321 1917 1
322 1918 1 ++
323 1919 1
324 1920 1 FUNCTIONAL DESCRIPTION:
325 1921 1
326 1922 1 This routine translates a field and the field sub-tree it owns.
327 1923 1
328 1924 1 FORMAL PARAMETERS:
329 1925 1
330 1926 1 FIELD_LOC - the location code of the entity describing the field.
331 1927 1 ROOT - true if this is the root
332 1928 1 FIRST_OFFSET - the offset, in bytes, of the byte following this
333 1929 1 field's prior sibling which shares the same
334 1930 1 immediate ancestor field. This is 0 for the first
335 1931 1 sub-field of a group.
336 1932 1 GROUP_OFFSET - the offset, in bytes, from the beginning of the record
337 1933 1 of the group containing this field.
338 1934 1
339 1935 1 IMPLICIT INPUTS:
340 1936 1
341 1937 1 NONE
342 1938 1
343 1939 1 IMPLICIT OUTPUTS:
344 1940 1
345 1941 1 NONE
346 1942 1
347 1943 1 ROUTINE VALUE:
348 1944 1
349 1945 1 The offset of the byte following the last byte mapped by the current
350 1946 1 field.
351 1947 1
352 1948 1 SIDE EFFECTS:
353 1949 1
354 1950 1 NONE
355 1951 1
356 1952 1 --
357 1953 2 BEGIN
358 1954 2 EXTERNAL REGISTER
359 1955 2 CTX = COM_REG_CTX: REF CTX_BLOCK,
360 1956 2 PRM = REG_PRM: REF VECTOR[2];
361 1957 2
362 1958 2 $GET_ATTS_LIST
363 1959 2 (FIELD_NAME, CDD$K_REC_NAME, STRING),
364 1960 2 (OFFSET, CDD$K_REC_OFFSET, NUMERIC),
365 1961 2 (LENGTH, CDD$K_REC_LENGTH, NUMERIC),
366 1962 2 (DATATYPE, CDD$K_REC_DATATYPE, NUMERIC),
367 1963 2 (BASE, CDD$K_REC_BASE, NUMERIC),
368 1964 2 (DIGITS, CDD$K_REC_DIGITS, NUMERIC),
369 1965 2 (SCALE, CDD$K_REC_SCALE, NUMERIC),
370 1966 2 (CHILDREN, CDD$K_REC_CHILDREN, ENTITY_LIST),
371 1967 2 (OVERLAYS, CDD$K_REC_OVERLAYS, ENTITY_LIST),
372 1968 2 (DIMENSION, CDD$K_REC_DIMENSION, ENTITY_LIST);
373 1969 2
374 1970 2 LOCAL
375 1971 2 CHILD_LOC,
376 1972 2 CHILD_OFFSET,
```



```

377      1973      2      LAST_OFFSET,
378      1974      2      STATUS;
379      1975      2
380      1976      2      $GET_ATTS_INIT;
381      1977      2
382      1978      2      ! Get the attributes we need for this routine
383      1979      2
384      1980      2      STATUS = $CDD$GET_ATTS(context = CTX[COM_CDD_CTX],
P 385      1981      2      location = FIELD_LOC,
386      1982      2      list = ATTR_LIST);
387      1983      2      IF NOT .STATUS THEN CRE_DIAG(SORS_CDDERROR, .STATUS);
388      1984      2      IF NOT .DATATYPE_STATUS THEN CRE_DIAG(SORS_BADPROTCL);
389      1985      2
390      1986      2      ! Ignore items of type VIRTUAL FIELD (Datatrieve COMPUTED BY).
391      1987      2      ! Ignore unnamed fields
392      1988      2
393      1989      2      IF .DATATYPE EQL CDD$K_REC_DTYPE_VIRT OR
394      1990      2      NOT .FIELD_NAME_STATUS
395      1991      2      THEN
396      1992      2      BEGIN
397      1993      2      STR$FREE1 DX(FIELD_NAME);
398      1994      2      RETURN .FIRST_OFFSET;
399      1995      2      END;
400      1996      2
401      1997      2      ! See if we're dealing with a REDEFINES clause...
402      1998      2
403      1999      2      IF .DATATYPE EQL CDD$K_REC_DTYPE_OVERLAY
404      2000      2      THEN
405      2001      2      IF NOT .OVERLAYS_STATUS
406      2002      2      THEN
407      2003      2      BEGIN
408      2004      2      CRE_DIAG(SORS_BADPROTCL);
409      2005      2      RETURN .FIRST_OFFSET;
410      2006      2      END
411      2007      2      ELSE
412      2008      2      RETURN CRE_OVERLAY_FIELDS(.OVERLAYS, .OVERLAYS_COUNT, .ROOT,
413      2009      2      .FIRST_OFFSET, .GROUP_OFFSET);
414      2010      2
415      2011      2
416      2012      2      ! It wasn't an overlay, so it must be a field.
417      2013      2
418      2014      2      ! Set up some defaults
419      2015      2
420      2016      2
421      2017      2      IF .BASE_STATUS AND .BASE NEQ 10
422      2018      2      THEN
423      2019      2      CRE_DIAG(SORS_ILLBASE);
424      2020      2
425      2021      2      IF NOT .SCALE_STATUS THEN SCALE = 0;
426      2022      2
427      2023      2      IF NOT .DIGITS_STATUS THEN DIGITS = 0;
428      2024      2
429      2025      2
430      2026      2      ! Check length and offset
431      2027      2
432      2028      2      IF NOT .OFFSET_STATUS
433      2029      2      THEN
```



```
434 2030 3 BEGIN
435 2031 3 OFFSET = .FIRST_OFFSET * 8;
436 2032 3 IF NOT .ROOT THEN CRE_DIAG(SORS$_MISLENOFF);
437 2033 3 END;
438 2034 3 IF (.OFFSET AND 7) NEQ 0
439 2035 3 THEN
440 2036 3 BEGIN
441 2037 3 OFFSET = (.OFFSET+7) AND NOT 7;
442 2038 3 CRE_DIAG(SORS$_BADLENOFF);
443 2039 3 END;
444 2040 3 IF NOT .LENGTH_STATUS OR .LENGTH EQL 0
445 2041 3 THEN
446 2042 3 BEGIN
447 2043 3 LENGTH = 8;
448 2044 3 CRE_DIAG(SORS$_MISLENOFF);
449 2045 3 END;
450 2046 3 IF (.LENGTH AND 7) NEQ 0
451 2047 3 THEN
452 2048 3 BEGIN
453 2049 3 LENGTH = (.LENGTH+7) AND NOT 7;
454 2050 3 CRE_DIAG(SORS$_BADLENOFF);
455 2051 3 END;
456 2052 3
457 2053 3 ! Convert length and offset from bits to bytes
458 2054 3 !
459 2055 3 LENGTH = .LENGTH / 8;
460 2056 3 OFFSET = .OFFSET / 8;
461 2057 3
462 2058 3
463 2059 3 ! Resolve the OCCURS clause and decide where this field ends.
464 2060 3 !
465 2061 3 IF NOT .DIMENSION_STATUS
466 2062 3 THEN
467 2063 3 LAST_OFFSET = .OFFSET + .LENGTH
468 2064 3 ELSE
469 2065 3 BEGIN
470 2066 3 IF .DIMENSION_COUNT NEQ 1
471 2067 3 THEN
472 2068 3 CRE_DIAG(SORS$_MULTIDIM);
473 2069 3 LAST_OFFSET = .OFFSET + .LENGTH * CRE_OCCURS_CLAUSE(.DIMENSION);
474 2070 3 END;
475 2071 3
476 2072 3
477 2073 3 ! Check the length.
478 2074 3 !
479 2075 3 IF (CASE .DATATYPE FROM 0 TO 35 OF
480 2076 3 SET
481 2077 3 [DSC$_K_DTYPE_Z,
482 2078 3 DSC$_K_DTYPE_T]: .LENGTH GTRU 'X'FFFF';
483 2079 3 [DSC$_K_DTYPE_BU,
484 2080 3 DSC$_K_DTYPE_B]: .LENGTH NEQ 1;
485 2081 3 [DSC$_K_DTYPE_WU,
486 2082 3 DSC$_K_DTYPE_W]: .LENGTH NEQ 2;
487 2083 3 [DSC$_K_DTYPE_LU,
488 2084 3 DSC$_K_DTYPE_L,
489 2085 3 DSC$_K_DTYPE_F]: .LENGTH NEQ 4;
490 2086 3 [DSC$_K_DTYPE_QU,
```



```

491 2087 3      DSC$K_DTYPE_Q,
492 2088 3      DSC$K_DTYPE_D,
493 2089 3      DSC$K_DTYPE_G,
494 2090 3      DSC$K_DTYPE_ADT]:      .LENGTH NEQ 8;
495 2091 3      [DSC$K_DTYPE_OU,
496 2092 3      DSC$K_DTYPE_O,
497 2093 3      DSC$K_DTYPE_H]:      .LENGTH NEQ 16;
498 2094 3      [DSC$K_DTYPE_P]:
499 2095 4      BEGIN
500 2096 4      IF .DIGITS STATUS
501 2097 4      THEN LENGTH = .DIGITS
502 2098 4      ELSE LENGTH = .LENGTH * 2 - 1;
503 2099 4      .LENGTH GTRU 31
504 2100 4      END;
505 2101 3      [DSC$K_DTYPE_NL,
506 2102 3      DSC$K_DTYPE_NR]:      .LENGTH GTRU 31;
507 2103 3      [DSC$K_DTYPE_NU,
508 2104 3      DSC$K_DTYPE_NLO,
509 2105 3      DSC$K_DTYPE_NRO,
510 2106 3      DSC$K_DTYPE_NZ]:      .LENGTH GTRU 31;
511 2107 3      [INRANGE,OUTRANGE]:
512 2108 4      BEGIN
513 2109 4      IF .DATATYPE EQL CDD$K_REC_DTYPE_DATE
514 2110 4      THEN
515 2111 4      .LENGTH NEQ 8
516 2112 4      ELSE
517 2113 5      BEGIN
518 2114 5      CRE_DIAG(SOR$ BADDDTYPE);
519 2115 5      DATATYPE = DSC$K_DTYPE_Z;
520 2116 5      .LENGTH GTRU 'X'FFFF'
521 2117 5      END
522 2118 4      END
523 2119 3      TES)
524 2120 3      THEN
525 2121 3      BEGIN
526 2122 3      IF .DIGITS STATUS
527 2123 3      THEN
528 2124 4      BEGIN
529 2125 4      CRE_DIAG(SOR$_INCDIGITS);
530 2126 4      DIGITS STATUS = CDD$_ATTNOTFND;
531 2127 3      END;
532 2128 3      END;
533 2129 3
534 2130 2      ! Check for an array
535 2131 2      !
536 2132 2      IF .DIMENSION STATUS
537 2133 2      THEN
538 2134 2      BEGIN
539 2135 2      STR$FREE1 DX(FIELD NAME);
540 2136 2      RETURN .LAST_OFFSET;
541 2137 2      END;
542 2138 2
543 2139 2
544 2140 2
545 2141 2      ! Finally...write out this field's description
546 2142 2      !
547 2143 2      SOR$$DEF_FIELD(
```



```
548 2144 2 FIELD_NAME,  
549 2145 2 .DATATYPE,  
550 2146 2 .GROUP_OFFSET+.OFFSET,  
551 2147 2 .LENGTH,  
552 2148 2 .SCALE,  
553 2149 2 .PRM[0],  
554 2150 2 .PRM[1];  
555 2151 2  
556 2152 2  
557 2153 2 ! Any children?  
558 2154 2  
559 2155 2 IF .CHILDREN_STATUS  
560 2156 2 THEN  
561 2157 2 BEGIN  
562 2158 2 CHILD_OFFSET = 0;  
563 2159 2 INCR I FROM 1 TO .CHILDREN_COUNT DO  
564 2160 2 BEGIN  
565 P 2161 4 STATUS = $CDD$GET_ENTITY_CELL(context = CTX[COM_CDD_CTX],  
566 P 2162 4 list = CHILDREN,  
567 P 2163 4 cell = .I,  
568 2164 4 location = CHILD_LOC);  
569 2165 4 IF NOT .STATUS  
570 2166 4 THEN  
571 2167 5 BEGIN  
572 2168 5 CRE_DIAG(SOR$BADPROTCL, .STATUS);  
573 2169 5 RETURN .FIRST_OFFSET;  
574 2170 4 END;  
575 2171 4 CHILD_OFFSET = CRE_FIELD(.CHILD_LOC, FALSE,  
576 2172 4 .CHILD_OFFSET, .GROUP_OFFSET+.OFFSET);  
577 2173 3 END;  
578 2174 2 END;  
579 2175 2  
580 2176 2  
581 2177 2 STR$FREE1_DX(FIELD_NAME);  
582 2178 2 RETURN .LAST_OFFSET;  
583 2179 2  
584 2180 1 END;
```

```
INFO#250 L1:1989  
Referenced LOCAL symbol DATATYPE is probably not initialized  
INFO#250 L1:2008  
Referenced LOCAL symbol OVERLAYS is probably not initialized  
INFO#250 L1:2017  
Referenced LOCAL symbol BASE is probably not initialized  
INFO#250 L1:2040  
Referenced LOCAL symbol LENGTH is probably not initialized  
INFO#250 L1:2069  
Referenced LOCAL symbol DIMENSION is probably not initialized
```

00000000	00000000	00000000	00000005	00010006	0000000A	00094	P.AAA:	.LONG	10, 65542, 5, 0, 0, 0, 65562, 4, 0	:
			00000000	00000004	0001001A	000AC				:
				0000	0004	000B8		.WORD	4, 0	:
	00000000	00000004	0001001B	00000000	000BC			.LONG	0, 65563, 4, 0	:
				0000	0004	000CC		.WORD	4, 0	:
	00000000	00000004	0001003A	00000000	000D0			.LONG	0, 65594, 4, 0	:
				0000	0004	000E0		.WORD	4, 0	:



00000000	00000004	0001001F	00000000	000E4	.LONG	0, 65567, 4, 0
			0000 0004	000F4	.WORD	4, 0
00000000	00000004	00010020	00000000	000F8	.LONG	0, 65568, 4, 0
			0000 0004	00108	.WORD	4, 0
00000000	00000004	00010021	00000000	0010C	.LONG	0, 65569, 4, 0
			0000 0004	0011C	.WORD	4, 0
00000000	00000002	00010039	00000000	00120	.LONG	0, 65593, 2, 0
			0000 0004	00130	.WORD	4, 0
00000000	00000002	00010036	00000000	00134	.LONG	0, 65590, 2, 0
			0000 0004	00144	.WORD	4, 0
00000000	00000002	00010024	00000000	00148	.LONG	0, 65572, 2, 0
			0000 0004	00158	.WORD	4, 0
			00000000	0015C	.LONG	0

.EXTRN CDD\$GET\_ATTS, CDD\$GET\_ENTITY\_CELL

				03FC 00000	CRE_FIELD:					
			59	00000000G	00	9E	00002	.WORD	Save R2,R3,R4,R5,R6,R7,R8,R9	1916
			58	FE93	CF	9E	00009	MOVAB	STR\$FREE1_DX, R9	
			5E	FF04	CE	9E	0000E	MOVAB	CRE_DIAG, R8	
28	AE	FF1A	CF	00CC	8F	28	00013	MOVAB	-252(SP), SP	
		FA	AD		0E	90	0001C	MOVAB	#204, P.AAA, ATTR_LIST	1968
		FB	AD		02	90	00020	MOVAB	#14, FIELD_NAME+2	1974
				F8	AD	B4	00024	MOVAB	#2, FIELD_NAME+3	
				FC	AD	D4	00027	CLRW	FIELD_NAME	
		3C	AE	F8	AD	9E	0002A	CLRL	FIELD_NAME+4	
		50	AE		6E	9E	0002F	MOVAB	FIELD_NAME, ATTR_LIST+20	
		64	AE	04	AE	9E	00033	MOVAB	OFFSET, ATTR_LIST+40	
		78	AE	08	AE	9E	00038	MOVAB	LENGTH, ATTR_LIST+60	
		90	AD	0C	AE	9E	0003D	MOVAB	DATATYPE, ATTR_LIST+80	
		A4	AD	10	AE	9E	00042	MOVAB	BASE, ATTR_LIST+100	
		B8	AD	14	AE	9E	00047	MOVAB	DIGITS, ATTR_LIST+120	
		CC	AD	24	AE	9E	0004C	MOVAB	SCALE, ATTR_LIST+140	
		E0	AD	18	AE	9E	00051	MOVAB	CHILDREN, ATTR_LIST+160	
		F4	AD	1C	AE	9E	00056	MOVAB	OVERLAYS, ATTR_LIST+180	
				28	AE	9F	0005B	MOVAB	DIMENSION, ATTR_LIST+200	
				04	AC	9F	0005E	PUSHAB	ATTR_LIST	1982
				0144	CB	9F	00061	PUSHAB	FIELD_LOC	
					03	FB	00065	PUSHAB	324(CTX)	
		00000000G	00		50	D0	0006C	CALLS	#3, CDD\$GET_ATTS	
			57		57	E8	0006F	MOVL	R0, STATUS	
			0B		57	DD	00072	BLBS	STATUS, 1\$	1983
				001C823C	8F	DD	00074	PUSHL	STATUS	
			68		02	FB	0007A	PUSHL	#1868348	
			09	70	AE	E8	0007D	CALLS	#2, CRE_DIAG	
				001C8234	8F	DD	00081	BLBS	DATATYPE_STATUS, 2\$	1984
			68		01	FB	00087	PUSHL	#1868340	
		00000101	8F	08	AE	D1	0008A	CALLS	#1, CRE_DIAG	
					04	13	00092	CMPL	DATATYPE, #257	1989
			08	34	AE	E8	00094	BEQL	3\$	
				F8	AD	9F	00098	BLBS	FIELD_NAME_STATUS, 4\$	1990
			69		01	FB	0009B	PUSHAB	FIELD_NAME	1993
					17	11	0009E	CALLS	#1, STR\$FREE1_DX	
		00000102	8F	08	AE	D1	000A0	BRB	5\$	1994
					24	12	000A8	CMPL	DATATYPE, #258	1999
			0C	D8	AD	E8	000AA	BNEQ	7\$	
								BLBS	OVERLAYS_STATUS, 6\$	2001



			001C8234	8F	DD	000AE	PUSHL	#1868340	2004
		68		01	FB	000B4	CALLS	#1, CRE_DIAG	
				01D1	31	000B7	BRW	37\$	2008
		7E	0C	AC	7D	000BA	MOVQ	FIRST_OFFSET, -(SP)	2009
			08	AC	DD	000BE	PUSHL	ROOT	2008
		7E	DE	AD	3C	000C1	MOVZWL	OVERLAYS_COUNT, -(SP)	
			28	AE	DD	000C5	PUSHL	OVERLAYS	
		0000V	CF	05	FB	000C8	CALLS	#5, CRE_OVERLAY_FIELDS	
					04	000CD	RET		
		0F	88	AD	E9	000CE	BLBC	BASE_STATUS, 8\$	2017
		0A	0C	AE	D1	000D2	CMPL	BASE, #10	
				09	13	000D6	BEQL	8\$	
			001C8242	8F	DD	000D8	PUSHL	#1868354	2019
		68		01	FB	000DE	CALLS	#1, CRE_DIAG	
		03	B0	AD	E8	000E1	BLBS	SCALE_STATUS, 9\$	2021
			14	AE	D4	000E5	CLRL	SCALE	
		03	9C	AD	E8	000E8	BLBS	DIGITS_STATUS, 10\$	2023
			10	AE	D4	000EC	CLRL	DIGITS	
		12	48	AE	E8	000EF	BLBS	OFFSET_STATUS, 11\$	2028
6E	0C	AC		03	78	000F3	ASHL	#3, FIRST_OFFSET, OFFSET	2031
		09	08	AC	E8	000F8	BLBS	ROOT, 11\$	2032
			001C8260	8F	DD	000FC	PUSHL	#1868384	
		68		01	FB	00102	CALLS	#1, CRE_DIAG	
		07		6E	93	00105	BITB	OFFSET, #7	2034
				11	13	00108	BEQL	12\$	
50		6E		07	C1	0010A	ADDL3	#7, OFFSET, R0	2037
6E		50		07	CB	0010E	BICL3	#7, R0, OFFSET	
			001C821A	8F	DD	00112	PUSHL	#1868314	2038
		68		01	FB	00118	CALLS	#1, CRE_DIAG	
		05	5C	AE	E9	0011B	BLBC	LENGTH_STATUS, 13\$	2040
			04	AE	D5	0011F	TSTL	LENGTH	
				0D	12	00122	BNEQ	14\$	
		04	AE	08	D0	00124	MOVL	#8, LENGTH	2043
			001C8260	8F	DD	00128	PUSHL	#1868384	2044
		68		01	FB	0012E	CALLS	#1, CRE_DIAG	
		07	04	AE	93	00131	BITB	LENGTH, #7	2046
				13	13	00135	BEQL	15\$	
50		04	AE	07	C1	00137	ADDL3	#7, LENGTH, R0	2049
04	AE	50		07	CB	0013C	BICL3	#7, R0, LENGTH	
			001C821A	8F	DD	00141	PUSHL	#1868314	2050
		68		01	FB	00147	CALLS	#1, CRE_DIAG	
		04	AE	08	C6	0014A	DIVL2	#8, LENGTH	2055
		6E		08	C6	0014E	DIVL2	#8, OFFSET	2056
		52	04	AE	D0	00151	MOVL	LENGTH, R2	2063
		06	EC	AD	E8	00155	BLBS	DIMENSION_STATUS, 16\$	2061
55		6E		52	C1	00159	ADDL3	R2, OFFSET, LAST_OFFSET	2063
				1E	11	0015D	BRB	18\$	
		01	F2	AD	B1	0015F	CMPW	DIMENSION_COUNT, #1	2066
				09	13	00163	BEQL	17\$	
			001C826A	8F	DD	00165	PUSHL	#1868394	2068
		68		01	FB	0016B	CALLS	#1, CRE_DIAG	
			1C	AE	DD	0016E	PUSHL	DIMENSION	2069
		0000V	CF	01	FB	00171	CALLS	#1, CRE_OCCURS_CLAUSE	
		55	50	52	C4	00176	MULL2	R2, R0	
		50		6E	C1	00179	ADDL3	OFFSET, R0, LAST_OFFSET	
		00	08	AE	CF	0017D	CASEL	DATATYPE, #0, #35	2075
0071	006C	0048	0063			00182	.WORD	23\$-19\$,-	



0071	006C	0052	0076	0018A	20\$-19\$,-		
0052	0076	0052	0076	00192	24\$-19\$,-		
009B	0063	0048	0048	0019A	25\$-19\$,-		
009B	009B	009B	009B	001A2	26\$-19\$,-		
0048	0048	0082	009B	001AA	21\$-19\$,-		
0052	007B	007B	0048	001B2	24\$-19\$,-		
0048	0048	0048	007B	001BA	25\$-19\$,-		
0052	0048	0048	0048	001C2	26\$-19\$,-		
					21\$-19\$,-		
					26\$-19\$,-		
					21\$-19\$,-		
					20\$-19\$,-		
					20\$-19\$,-		
					23\$-19\$,-		
					32\$-19\$,-		
					32\$-19\$,-		
					32\$-19\$,-		
					32\$-19\$,-		
					32\$-19\$,-		
					29\$-19\$,-		
					20\$-19\$,-		
					20\$-19\$,-		
					20\$-19\$,-		
					27\$-19\$,-		
					27\$-19\$,-		
					21\$-19\$,-		
					27\$-19\$,-		
					20\$-19\$,-		
					20\$-19\$,-		
					20\$-19\$,-		
					20\$-19\$,-		
					20\$-19\$,-		
					21\$-19\$		
00000100	8F	08	AE	D1 001CA	20\$:	CMPL DATATYPE, #256	2109
			05	12 001D2		BNEQ	
	08		52	D1 001D4	21\$:	CMPL	
			27	11 001D7		BRB	
	001C8212		8F	DD 001D9	22\$:	PUSHL	
	68		01	FB 001DF		CALLS	
		08	AE	D4 001E2		CLRL	
0000FFFF	8F		52	D1 001E5	23\$:	CMPL	
			32	11 001EC		BRB	
	01		52	D1 001EE	24\$:	CMPL	
			0D	11 001F1		BRB	
	02		52	D1 001F3	25\$:	CMPL	
			08	11 001F3		BRB	
	04		52	D1 001F8	26\$:	CMPL	
			03	11 001FB		BRB	
	10		52	D1 001FD	27\$:	CMPL	
			35	13 00200	28\$:	BEQL	
			1E	11 00202		BRB	
	07	9C	AD	E9 00204	29\$:	BLBC	
	04	10	AE	D0 00208		MOVL	
			08	11 0020D		BRB	
04	AE		01	78 0020F	30\$:	ASHL	
						#1, R2, LENGTH	2098



		04	AE	D7	00214	DECL	LENGTH	:		
	1F	04	AE	D1	00217	31\$:	CMPL	LENGTH, #31	: 2099	
			03	11	0021B		BRB	33\$	:	
	1F		52	D1	0021D	32\$:	CMPL	R2, #31	: 2106	
			15	1B	00220	33\$:	BLEQU	35\$	:	
	11	9C	AD	E9	00222	34\$:	BLBC	DIGITS_STATUS, 35\$	: 2122	
	68	001C825A	8F	DD	00226		PUSHL	#1868378	: 2125	
			01	FB	0022C		CALLS	#1, CRE_DIAG	:	
	9C	AD	EF	D0	0022F		MOVL	CDD_ATTNOTFND, DIGITS_STATUS	: 2126	
	6A	00000000	AD	E8	00237	35\$:	BLBS	DIMENSION_STATUS, 40\$	: 2133	
	7E	EC	6A	7D	0023B		MOVQ	(PRM), -(SP)	: 2149	
			1C	AE	DD	0023E	PUSHL	SCALE	: 2148	
			10	AE	DD	00241	PUSHL	LENGTH	: 2147	
53	10	AC	10	AE	C1	00244	ADDL3	OFFSET, GROUP_OFFSET, R3	: 2146	
			53	DD	0024A		PUSHL	R3	:	
			1C	AE	DD	0024C	PUSHL	DATATYPE	: 2145	
			F8	AD	9F	0024F	PUSHAB	FIELD_NAME	: 2143	
00000000G	00		07	FB	00252		CALLS	#7, SOR\$\$DEF_FIELD	:	
	48	C4	AD	E9	00259		BLBC	CHILDREN_STATUS, 40\$	: 2155	
			56	D4	0025D		CLRL	CHILD_OFFSET	: 2158	
	54	CA	AD	3C	0025F		MOVZWL	CHILDREN_COUNT, R4	: 2159	
			52	D4	00263		CLRL	I	:	
			3A	11	00265		BRB	39\$	:	
			20	AE	9F	00267	36\$:	PUSHAB	CHILD_LOC	: 2164
			52	DD	0026A		PUSHL	I	:	
			2C	AE	9F	0026C	PUSHAB	CHILDREN	:	
		0144	CB	9F	0026F		PUSHAB	324(CTX)	:	
00000000G	00		04	FB	00273		CALLS	#4, CDD\$GET_ENTITY_CELL	:	
	57		50	D0	0027A		MOVL	R0, STATUS	:	
	10		57	E8	0027D		BLBS	STATUS, 38\$	: 2165	
			57	DD	00280		PUSHL	STATUS	: 2168	
		001C8234	8F	DD	00282		PUSHL	#1868340	:	
	68		02	FB	00288		CALLS	#2, CRE_DIAG	:	
	50	0C	AC	D0	0028B	37\$:	MOVL	FIRST_OFFSET, R0	: 2169	
				04	0028F		RET		:	
			53	DD	00290	38\$:	PUSHL	R3	: 2172	
			56	DD	00292		PUSHL	CHILD_OFFSET	:	
			7E	D4	00294		CLRL	-(SP)	: 2171	
			2C	AE	DD	00296	PUSHL	CHILD_LOC	:	
	0160	C8	04	FB	00299		CALLS	#4, CRE_FIELD	:	
		56	50	D0	0029E		MOVL	R0, CHILD_OFFSET	:	
C2		52	54	F3	002A1	39\$:	AOBLEQ	R4, I, 36\$	: 2159	
			AD	9F	002A5	40\$:	PUSHAB	FIELD_NAME	: 2177	
	69	F8	01	FB	002AB		CALLS	#1, STR\$FREE1_DX	:	
	50		55	D0	002AB		MOVL	LAST_OFFSET, R0	: 2178	
			04	002AE			RET		: 2180	

; Routine Size: 687 bytes, Routine Base: SOR\$R0\_CODE + 0160



```
586 2181 1 ROUTINE CRE_OCCURS_CLAUSE(DIMENSION_LOC): CAL_CTXREG =
587 2182 1
588 2183 1 ++
589 2184 1
590 2185 1 FUNCTIONAL DESCRIPTION:
591 2186 1
592 2187 1 This routine generates an OCCURS clause of the form:
593 2188 1
594 2189 1 OCCURS n [TO n] TIMES [DEPENDING ON name]
595 2190 1
596 2191 1 FORMAL PARAMETERS:
597 2192 1
598 2193 1 DIMENSION_LOC - the location code of the CDD$K_REC_DIMENSION list.
599 2194 1
600 2195 1 IMPLICIT INPUTS:
601 2196 1
602 2197 1 NONE
603 2198 1
604 2199 1 IMPLICIT OUTPUTS:
605 2200 1
606 2201 1 NONE
607 2202 1
608 2203 1 RETURN VALUE:
609 2204 1
610 2205 1 The number of item occurrences.
611 2206 1
612 2207 1 SIDE EFFECTS:
613 2208 1
614 2209 1 NONE
615 2210 1
616 2211 1 --
617 2212 2 BEGIN
618 2213 2 EXTERNAL REGISTER
619 2214 2 CTX = COM_REG_CTX: REF CTX_BLOCK;
620 2215 2
621 P 2216 2 $GET_ATTS_LIST
622 P 2217 2 (UPPER_BOUND, CDD$K_REC_UPPER_BOUND, NUMERIC),
623 P 2218 2 (LOWER_BOUND, CDD$K_REC_LOWER_BOUND, NUMERIC),
624 P 2219 2 (STRIDE, CDD$K_REC_STRIDE, NUMERIC),
625 P 2220 2 (MIN_OCCURS, CDD$K_REC_MIN_OCCURS, NUMERIC),
626 2221 2 (MAX_OCCURS, CDD$K_REC_MAX_OCCURS, NUMERIC));
627 2222 2
628 2223 2 LOCAL
629 2224 2 DIMENSION,
630 2225 2 OCCURS,
631 2226 2 STATUS;
632 2227 2
633 2228 2 $GET_ATTS_INIT;
634 2229 2
635 2230 2 ! Get the dimension attributes
636 2231 2
637 P 2232 2 STATUS = $CDD$GET_ENTITY_CELL(context = CTX[COM_CDD_CTX],
638 P 2233 2 list = DIMENSION_LOC,
639 P 2234 2 cell = 1,
640 2235 2 location = DIMENSION);
641 2236 2 IF NOT .STATUS
642 2237 2 THEN
```



```

: 643      2238      3      BEGIN
: 644      2239      3      CRE_DIAG(SOR$_CDDERROR, .STATUS);
: 645      2240      3      RETURN 1;                                ! Pretend it occurs only once
: 646      2241      2      END;
: 647      2242      2
: 648      P 2243      2      STATUS = $CDD$GET_ATT$ (context = CTX[COM_CDD_CTX],
: 649      P 2244      2      location = DIMENSION,
: 650      2245      2      list = ATTR_LIST);
: 651      2246      2      IF NOT .STATUS
: 652      2247      2      THEN
: 653      2248      3      BEGIN
: 654      2249      3      CRE_DIAG(SOR$_CDDERROR, .STATUS);
: 655      2250      3      RETURN 1;
: 656      2251      2      END;
: 657      2252      2
: 658      2253      2
: 659      2254      2      IF .MAX_OCCURS_STATUS
: 660      2255      2      THEN
: 661      2256      2      OCCURS = .MAX_OCCURS
: 662      2257      2      ELIF .UPPER_BOUND_STATUS
: 663      2258      2      THEN
: 664      2259      3      BEGIN
: 665      2260      3      IF NOT .LOWER_BOUND_STATUS THEN LOWER_BOUND = 1;
: 666      2261      3      OCCURS = .UPPER_BOUND - .LOWER_BOUND + 1;
: 667      2262      3      END
: 668      2263      2      ELSE
: 669      2264      2      OCCURS = 0;
: 670      2265      2
: 671      2266      2      IF .OCCURS LEQ 0
: 672      2267      2      THEN
: 673      2268      3      BEGIN
: 674      2269      3      CRE_DIAG(SOR$_BADOCCURS);
: 675      2270      3      OCCURS = 1;
: 676      2271      2      END;
: 677      2272      2
: 678      2273      2      IF NOT .STRIDE_STATUS
: 679      2274      2      THEN
: 680      2275      2      CRE_DIAG(SOR$_BADPROTCL, .STATUS);
: 681      2276      2
: 682      2277      2      RETURN .OCCURS;
: 683      2278      1      END;

```

```

: INFO#250      L1:2256
: Referenced LOCAL symbol MAX_OCCURS is probably not initialized
: INFO#250      L1:2261
: Referenced LOCAL symbol UPPER_BOUND is probably not initialized

```

00000000	00000004	00010045	00000005	0040F	P.AAB:	.BLKB	1	
			0000 0004	00410		.LONG	5,	65605, 4, 0
				00420		.WORD	4,	0
00000000	00000004	00010044	00000000	00424		.LONG	0,	65604, 4, 0
			0000 0004	00434		.WORD	4,	0
00000000	00000004	00010026	00000000	00438		.LONG	0,	65574, 4, 0
			0000 0004	00448		.WORD	4,	0
00000000	00000004	00010027	00000000	0044C		.LONG	0,	65575, 4, 0
			0000 0004	0045C		.WORD	4,	0



00000000 00000004 00010025 00000000 00460  
0000 0004 00470  
00000000 00474

.LONG 0, 65573, 4, 0  
.WORD 4, 0  
.LONG 0

..  
..  
..

## 007C 00000 CRE\_OCCURS\_CLAUSE:

18	AE	87	56	FB82	CF	9E	00002	.WORD	Save R2,R3,R4,R5,R6	2181
		2C	5E	80	AE	9E	00007	MOVAB	CRE_DIAG, R6	
		40	AF	0068	8F	28	0000B	MOVAB	-128(SP), SP	2221
		54	AE		6E	9E	00013	MOVAB	#104, P.AAB, ATTR_LIST	2226
		68	AE	04	AE	9E	00017	MOVAB	UPPER_BOUND, ATTR_LIST+20	
		7C	AE	08	AE	9E	0001C	MOVAB	LOWER_BOUND, ATTR_LIST+40	
			AE	0C	AE	9E	00021	MOVAB	STRIDE, ATTR_LIST+60	
			AE	10	AE	9E	00026	MOVAB	MIN_OCCURS, ATTR_LIST+80	
			AE	14	AE	9F	0002B	MOVAB	MAX_OCCURS, ATTR_LIST+100	
					01	DD	0002E	PUSHAB	DIMENSION	2235
				04	AC	9F	00030	PUSHL	#1	
				0144	CB	9F	00033	PUSHAB	DIMENSION_LOC	
			00		04	FB	00037	PUSHAB	324(CTX)	
			53		50	D0	0003E	CALLS	#4, CDD\$GET_ENTITY_CELL	
			17		53	E9	00041	MOVL	R0, STATUS	
				18	AE	9F	00044	BLBC	STATUS, 1\$	2236
				18	AE	9F	00047	PUSHAB	ATTR_LIST	2245
				0144	CB	9F	0004A	PUSHAB	DIMENSION	
			00		03	FB	0004E	PUSHAB	324(CTX)	
			53		50	D0	00055	CALLS	#3, CDD\$GET_ATTS	
			OF		53	E8	00058	MOVL	R0, STATUS	
					53	DD	0005B	BLBS	STATUS, 2\$	2246
				001C823C	8F	DD	0005D	PUSHL	STATUS	2249
			66		02	FB	00063	PUSHL	#1868348	
			50		01	D0	00066	CALLS	#2, CRE_DIAG	
						04	00069	MOVL	#1, R0	2250
			06	74	AE	E9	0006A	RET		
			52	10	AE	D0	0006E	BLBC	MAX_OCCURS_STATUS, 3\$	2254
					19	11	00072	MOVL	MAX_OCCURS, OCCURS	2256
			13	24	AE	E9	00074	BRB	6\$	
			04	38	AE	E8	00078	BLBC	UPPER_BOUND_STATUS, 5\$	2257
			AE		01	D0	0007C	BLBS	LOWER_BOUND_STATUS, 4\$	2260
50	04		6E	04	AE	C3	00080	MOVL	#1, LOWER_BOUND	
			52	01	A0	9E	00085	SUBL3	LOWER_BOUND, UPPER_BOUND, R0	2261
					02	11	00089	MOVAB	1(R0), OCCURS	
					52	D4	0008B	BRB	6\$	2256
					0C	14	0008D	CLRL	OCCURS	2264
				001C8224	8F	DD	0008F	BGTR	7\$	2266
			66		01	FB	00095	PUSHL	#1868324	2269
			52		01	D0	00098	CALLS	#1, CRE_DIAG	
			0B	4C	AE	E8	0009B	MOVL	#1, OCCURS	2270
					53	DD	0009F	BLBS	STRIDE_STATUS, 8\$	2273
				001C8234	8F	DD	000A1	PUSHL	STATUS	2275
			66		02	FB	000A7	PUSHL	#1868340	
			50		52	D0	000AA	CALLS	#2, CRE_DIAG	
					04	000AD		MOVL	OCCURS, -R0	2277
								RET		2278

; Routine Size: 174 bytes, Routine Base: SOR\$RO\_CODE + 0478



CRE\_TRANS  
V04=000

User Literal Definitions

D 3  
16-Sep-1984 01:08:56  
14-Sep-1984 13:10:31

VAX-11 Bliss-32 V4.0-742  
[SORT32.SRC]CRETRANS.B32;1

Page 22  
(8)

CR  
VO



## User Literal Definitions

```
: 685 2279 1 ROUTINE CRE_OVERLAY_FIELDS(OVERLAYS_LOC, OVERLAY_COUNT, ROOT, FIRST_OFFSET,
: 686 2280 1     GROUP_OFFSET)
: 687 2281 1     : CAL_CTXREG_LB =
: 688 2282 1
: 689 2283 1 ++
: 690 2284 1
: 691 2285 1 FUNCTIONAL DESCRIPTION:
: 692 2286 1
: 693 2287 1     This routine handles REDEFINES clauses. It forces the
: 694 2288 1     source to look like:
: 695 2289 1
: 696 2290 1     level name1 ...
: 697 2291 1
: 698 2292 1     level name2 REDEFINES name1 ...
: 699 2293 1     .
: 700 2294 1     .
: 701 2295 1     level namei REDEFINES name1 ...
: 702 2296 1
: 703 2297 1 FORMAL PARAMETERS:
: 704 2298 1
: 705 2299 1     OVERLAYS_LOC - the location code of the entity list describing the
: 706 2300 1     overlays.
: 707 2301 1     OVERLAY_COUNT - the number of overlays.
: 708 2302 1     ROOT - true if this is the root
: 709 2303 1     FIRST_OFFSET - the offset, in bytes, of the byte following this
: 710 2304 1     field's prior sibling which shares the same
: 711 2305 1     immediate ancestor field. This is 0 for the first
: 712 2306 1     sub-field of a group.
: 713 2307 1     GROUP_OFFSET - the offset, in bytes, from the beginning of the record
: 714 2308 1     of the group containing this field.
: 715 2309 1
: 716 2310 1 IMPLICIT INPUTS:
: 717 2311 1
: 718 2312 1     NONE
: 719 2313 1
: 720 2314 1 IMPLICIT OUTPUTS:
: 721 2315 1
: 722 2316 1     NONE
: 723 2317 1
: 724 2318 1 RETURN VALUE:
: 725 2319 1
: 726 2320 1     The offset of the byte following the last byte mapped by this field.
: 727 2321 1
: 728 2322 1 SIDE EFFECTS:
: 729 2323 1
: 730 2324 1     NONE
: 731 2325 1
: 732 2326 1 --
: 733 2327 2 BEGIN
: 734 2328 2 EXTERNAL REGISTER
: 735 2329 2     CTX = COM_REG_CTX: REF CTX_BLOCK,
: 736 2330 2     PRM = REG_PRM: REF VECTOR[2];
: 737 2331 2 MAP
: 738 2332 2     OVERLAY_COUNT: WORD;
: 739 2333 2
: 740 2334 2 LOCAL
: 741 2335 2     FIELD_NAME: $DSC VOLATILE,
```



```

742      2336 2      LAST_OFFSET,
743      2337 2      OVERLAY_LOC,
744      2338 2      OVERLAY_SIZE:      WORD,
745      2339 2      STATUS;
746      2340 2
747      2341 2      CRE_DECLARE_HANDLER(FIELD_NAME);
748      2342 2
749      2343 2      LAST_OFFSET = .FIRST_OFFSET;
750      2344 2      INCR I FROM 1 TO .OVERLAY_COUNT DO
751      2345 3      BEGIN
752      P 2346 3      STATUS = $CDD$GET_ENTITY_CELL(context = CTX[COM_CDD_CTX],
753      P 2347 3      list = OVERLAY_LOC,
754      P 2348 3      cell = .I,
755      2349 3      location = OVERLAY_LOC);
756      2350 3      IF NOT .STATUS
757      2351 3      THEN
758      2352 4      BEGIN
759      2353 4      CRE_DIAG(SORS_BADPROTCL, .STATUS);
760      2354 4      RETURN .FIRST_OFFSET;
761      2355 3      END;
762      2356 3
763      P 2357 3      STATUS = $CDD$GET_ENTITY_LIST_ATT(context = CTX[COM_CDD_CTX],
764      P 2358 3      entity = OVERLAY_LOC,
765      P 2359 3      attribute = CDD$R_REC_FIELDS,
766      P 2360 3      location = OVERLAY_LOC,
767      2361 3      list_size = OVERLAY_SIZE);
768      2362 3      IF NOT .STATUS
769      2363 3      THEN
770      2364 4      BEGIN
771      2365 4      CRE_DIAG(SORS_BADPROTCL, .STATUS);
772      2366 4      RETURN .FIRST_OFFSET;
773      2367 3      END;
774      2368 3      IF .OVERLAY_SIZE NEQ 1
775      2369 3      THEN
776      2370 3      CRE_DIAG(SORS_BADOVRLAY);
777      2371 3
778      P 2372 3      STATUS = $CDD$GET_ENTITY_CELL(context = CTX[COM_CDD_CTX],
779      P 2373 3      list = OVERLAY_LOC,
780      P 2374 3      cell = 1,
781      2375 3      location = OVERLAY_LOC);
782      2376 3      IF NOT .STATUS
783      2377 3      THEN
784      2378 4      BEGIN
785      2379 4      CRE_DIAG(SORS_BADPROTCL, .STATUS);
786      2380 4      RETURN .FIRST_OFFSET;
787      2381 3      END;
788      2382 3      IF .I EQL 1
789      2383 3      THEN
790      2384 4      BEGIN
791      2385 4      ! Get the name and create the name string.
792      2386 4      !
793      P 2387 4      STATUS = $CDD$GET_STRING_ATT(context = CTX[COM_CDD_CTX],
794      P 2388 4      entity = OVERLAY_LOC,
795      P 2389 4      attribute = CDD$R_REC_NAME,
796      2390 4      value = FIELD_NAME);
797      2391 4      IF NOT .STATUS THEN CRE_DIAG(SORS_BADPROTCL, .STATUS);
798      2392 3      END;
```



799	2393	3	
800	2394	3	LAST_OFFSET = MAXU(
801	2395	3	.LAST_OFFSET,
802	2396	3	CRE_FIELD(.OVERLAY_LOC, .ROOT, .FIRST_OFFSET, .GROUP_OFFSET));
803	2397	2	END;
804	2398	2	
805	2399	2	STR\$FREE1_DX(FIELD_NAME);
806	2400	2	
807	2401	2	RETURN .LAST_OFFSET;
808	2402	1	END;

```
.EXTRN CDD$GET_ENTITY_LIST_ATT
.EXTRN CDD$GET_STRING_ATT
```

01FC 00000 CRE_OVERLAY FIELDS:						
58	00000000G	00	9E 00002	WORD	Save R2,R3,R4,R5,R6,R7,R8	2279
57	FACD	CF	9E 00009	MOVAB	CDD\$GET_ENTITY_CELL, R8	
5E		10	C2 0000E	MOVAB	CRE_DIAG, R7	
	08	AE	7C 00011	SUBL2	#16, SP	
6D	00E4	CF	DE 00014	CLRQ	FIELD_NAME	2327
OA		0E	90 00019	MOVAL	8\$, (FP)	
OB		02	90 0001D	MOVB	#14, FIELD_NAME+2	2341
	08	AE	B4 00021	MOVB	#2, FIELD_NAME+3	
	0C	AE	D4 00024	CLRW	FIELD_NAME	
54	10	AC	DO 00027	CLRL	FIELD_NAME+4	
52		54	DO 0002B	MOVL	FIRST_OFFSET, R4	2343
56	08	AC	3C 0002E	MOVL	R4, LAST_OFFSET	
		55	D4 00032	MOVZWL	OVERLAY_COUNT, R6	2344
		00B1	31 00034	CLRL	I	2396
	04	AE	9F 00037	BRW	7\$	
		55	DD 0003A	PUSHAB	OVERLAY_LOC	2349
	04	AC	9F 0003C	PUSHL	I	
	0144	CB	9F 0003F	PUSHAB	OVERLAYS_LOC	
68		04	FB 00043	PUSHAB	324(CTX)	
53		50	DO 00046	CALLS	#4, CDD\$GET_ENTITY_CELL	
42		53	E9 00049	MOVL	R0, STATUS	
		5E	DD 0004C	BLBC	STATUS, 3\$	2350
	08	AE	9F 0004E	PUSHL	SP	2361
	00010004	8F	DD 00051	PUSHAB	OVERLAY_LOC	
	10	AE	9F 00057	PUSHL	#65540	
	0144	CB	9F 0005A	PUSHAB	OVERLAY_LOC	
00000000G	00	05	FB 0005E	PUSHAB	324(CTX)	
	53	50	DO 00065	CALLS	#5, CDD\$GET_ENTITY_LIST_ATT	
	23	53	E9 00068	MOVL	R0, STATUS	
	01	6E	B1 0006B	BLBC	STATUS, 3\$	2362
		09	13 0006E	CMPW	OVERLAY_SIZE, #1	2368
	001C822A	8F	DD 00070	BEQL	2\$	
67		01	FB 00076	PUSHL	#1868330	2370
	04	AE	9F 00079	CALLS	#1, CRE_DIAG	
		01	DD 0007C	PUSHAB	OVERLAY_LOC	2375
	0C	AE	9F 0007E	PUSHL	#1	
	0144	CB	9F 00081	PUSHAB	OVERLAY_LOC	
68		04	FB 00085	PUSHAB	324(CTX)	
53		50	DO 00088	CALLS	#4, CDD\$GET_ENTITY_CELL	
				MOVL	R0, STATUS	



0F		53	E8	0008B	BLBS	STATUS, 4\$	: 2376	
		53	DD	0008E	3\$: PUSHL	STATUS	: 2379	
	001C8234	8F	DD	00090	PUSHL	#1868340		
67		02	FB	00096	CALLS	#2, CRE_DIAG		
50		54	DD	00099	MOVL	R4, R0	: 2380	
			04	0009C	RET			
01		55	D1	0009D	4\$: CMPL	I, #1	: 2382	
		28	12	000A0	BNEQ	5\$		
	08	AE	9F	000A2	PUSHAB	FIELD_NAME	: 2390	
	00010006	8F	DD	000A5	PUSHL	#65542		
	0C	AE	9F	000AB	PUSHAB	OVERLAY_LOC		
	0144	CB	9F	000AE	PUSHAB	324(CTX)		
00000000G	00	04	FB	000B2	CALLS	#4, CDD\$GET_STRING_ATT		
	53	50	DD	000B9	MOVL	R0, STATUS		
	0B	53	E8	000BC	BLBS	STATUS, 5\$	: 2391	
		53	DD	000BF	PUSHL	STATUS		
	001C8234	8F	DD	000C1	PUSHL	#1868340		
		02	FB	000C7	CALLS	#2, CRE_DIAG		
	14	AC	DD	000CA	5\$: PUSHL	GROUP_OFFSET	: 2396	
		54	DD	000CD	PUSHL	R4		
	0C	AC	DD	000CF	PUSHL	ROOT		
	10	AE	DD	000D2	PUSHL	OVERLAY_LOC		
0160	C7	04	FB	000D5	CALLS	#4, CRE_FIELD		
	51	52	DD	000DA	MOVL	LAST_OFFSET, R1		
	50	51	D1	000DD	CMPL	R1, R0		
		03	1E	000E0	BGEQU	6\$		
	51	50	DD	000E2	MOVL	R0, R1		
	52	51	DD	000E5	6\$: MOVL	R1, LAST_OFFSET	: 2394	
FF49	55	01	56	F1	7\$: ACBL	R6, #1, I, 1\$	: 2344	
			AE	9F	PUSHAB	FIELD_NAME	: 2399	
	00000000G	00	01	FB	000F1	CALLS	#1, STR\$FREE1_DX	
		50	52	DD	000F8	MOVL	LAST_OFFSET, R0	: 2401
			04	000FB	RET		: 2402	
			0000	000FC	8\$: .WORD	Save nothing	: 2327	
	50	08	AC	DD	000FE	MOVL	8(AP), R0	
	50	04	A0	DD	00102	MOVL	4(R0), R0	
		F8	A0	9F	00106	PUSHAB	FIELD_NAME	
			01	DD	00109	PUSHL	#1	
			5E	DD	0010B	PUSHL	SP	
	7E	04	AC	7D	0010D	MOVQ	4(AP), -(SP)	
FA26	CF	03	FB	00111	CALLS	#3, FREE_DYN		
		04	00116	RET				

; Routine Size: 279 bytes, Routine Base: SOR\$RO\_CODE + 0526



```

: 810      2403 1 ROUTINE CRE_TRANS_RECORD(RECORD_LOC): CAL_CTXREG_LB NOVALUE =
: 811      2404 1
: 812      2405 1 !++
: 813      2406 1
: 814      2407 1 FUNCTIONAL DESCRIPTION:
: 815      2408 1
: 816      2409 1     This routine accepts the location code of a record description and
: 817      2410 1     translates the record to COBOL source text.
: 818      2411 1
: 819      2412 1     If the record cannot be translated, a diagnostic is issued.
: 820      2413 1
: 821      2414 1 FORMAL PARAMETERS:
: 822      2415 1
: 823      2416 1     RECORD_LOC      - the location code of the record's terminal node.
: 824      2417 1
: 825      2418 1 IMPLICIT INPUTS:
: 826      2419 1
: 827      2420 1     CTX[COM_CDD_CTX]      - the CDD stream ID for the stream we're using
: 828      2421 1
: 829      2422 1 IMPLICIT OUTPUTS:
: 830      2423 1
: 831      2424 1     NONE
: 832      2425 1
: 833      2426 1 RETURN VALUE:
: 834      2427 1
: 835      2428 1     NONE
: 836      2429 1
: 837      2430 1 SIDE EFFECTS:
: 838      2431 1
: 839      2432 1     NONE
: 840      2433 1
: 841      2434 1 --
: 842      2435 2 BEGIN
: 843      2436 2 EXTERNAL REGISTER
: 844      2437 2     CTX = COM_REG_CTX:    REF CTX_BLOCK,
: 845      2438 2     PRM = REG_PRM:      REF VECTOR[2];
: 846      2439 2
: 847      2440 2 $GET_ATTS_LIST
: 848      2441 2     (SHORT_PATH, CDD$K_SHORT_PATHNAME, STRING),
: 849      2442 2     (PATHNAME, CDD$K_PATHNAME, STRING),
: 850      2443 2     (ROOT, CDD$K_REC_ROOT, ENTITY),
: 851      2444 2     (FORMAT, CDD$K_REC_FORMAT, NUMERIC),
: 852      2445 2     (LITERALS, CDD$K_REC_LITERALS, ENTITY_LIST));
: 853      2446 2 LOCAL
: 854      2447 2 STATUS;
: 855      2448 2
: 856      2449 2 CRE_DECLARE_HANDLER(PATHNAME,SHORT_PATH);
: 857      2450 2 $GET_ATTS_INIT;
: 858      2451 2
: 859      2452 2 ! Fetch attributes used in this routine
: 860      2453 2 !
: 861      2454 2 !
: 862      2455 2 STATUS = $CDD$GET_ATTS(context = CTX[COM_CDD_CTX],
: 863      2456 2     location = RECORD_LOC,
: 864      2457 2     list = ATTR_LIST);
: 865      2458 2 IF NOT .STATUS
: 866      2459 2 THEN
```



```
: 867      2460  2      RETURN CRE_DIAG(SOR$ CDDERROR, .STATUS);
: 868      2461  2      IF NOT .SHORT_PATH_STATUS
: 869      2462  2      THEN
: 870      2463  2      RETURN CRE_DIAG(SOR$ BADPROTCL, .SHORT_PATH_STATUS);
: 871      2464  2      CTX[COM_CDD_PATH] = SHORT_PATH;      ! Set the address of the descriptor
: 872      2465  2      IF NOT .PATHNAME_STATUS
: 873      2466  2      THEN
: 874      2467  2      RETURN CRE_DIAG(SOR$ BADPROTCL, .PATHNAME_STATUS);
: 875      2468  2      IF
: 876      2469  2      NOT .FORMAT_STATUS OR
: 877      2470  2      .FORMAT NEQ CDD$K_REC_FIXED AND
: 878      2471  2      .FORMAT NEQ CDD$K_REC_VARIABLE
: 879      2472  2      THEN
: 880      2473  2      RETURN CRE_DIAG(SOR$ BADPROTCL, .PATHNAME_STATUS);
: 881      2474  2      IF .LITERALS_STATUS NEQ CDD$ ATTNOTFND
: 882      2475  2      THEN
: 883      2476  2      RETURN CRE_DIAG(SOR$ ILLLITERL);
: 884      2477  2
: 885      2478  2      ! Translate the record
: 886      2479  2      !
: 887      2480  2      IF .ROOT_STATUS
: 888      2481  2      THEN
: 889      2482  2      CRE_FIELD(.ROOT, TRUE, 0, 0)
: 890      2483  2      ELSE
: 891      2484  2      CRE_DIAG(SOR$ BADLENOFF);
: 892      2485  2
: 893      2486  2      STR$FREE1_DX(PATHNAME);
: 894      2487  2      STR$FREE1_DX(SHORT_PATH);
: 895      2488  1      END;
```

```
: INFO#250      L1:2470
: Referenced LOCAL symbol FORMAT is probably not initialized
: INFO#250      L1:2482
: Referenced LOCAL symbol ROOT is probably not initialized
```

00000000	00000000	00000000	00000005	18000008	00000005	0063D	P.AAC:	.BLKB	3	
0001003B	00000000	00000000	00000000	00000005	18000007	00640		.LONG	5,	402653192, 5, 0, 0, 0, 402653191, 5, -
				00000000	00000001	00658				0, 0, 0, 65595, 1, 0
				0000 0004	00670					
				00000000	00678			.WORD	4, 0	
	00000000	00000004	0001003C	00000000	0067C			.LONG	0, 65596, 4, 0	
				0000 0004	0068C			.WORD	4, 0	
	00000000	00000002	00010005	00000000	00690			.LONG	0, 65541, 2, 0	
				0000 0004	006A0			.WORD	4, 0	
				00000000	006A4			.LONG	0	

## 00FC 0000 CRE\_TRANS RECORD:

			57	00000000G	00	9E	00002	.WORD	Save R2,R3,R4,R5,R6,R7	: 2403
			56	F94B	CF	9E	00009	MOVAB	STR\$FREE1_DX, R7	:
			5E	FF7C	CE	9E	0000E	MOVAB	CRE_DIAG, R6	:
								MOVAB	-132(SP), SP	:
OC	AE	FF7E	CF	0068	8F	28	00013	MOVAB	#104, P.AAC, ATTR_LIST	: 2445
				7C	AE	7C	0001C	CLRQ	SHORT_PATH	:
				74	AE	7C	0001F	CLRQ	PATHNAME	:



## User Literal Definitions

K 3  
16-Sep-1984 01:08:56  
14-Sep-1984 13:10:31VAX-11 Bliss-32 V4.0-742  
[SORT32.SRC]CRETRANS.B32;1Page 29  
(10)

	6D	00DB	CF	DE	00022	MOVAL	10\$, (FP)	
76	AE		0E	90	00027	MOVB	#14, PATHNAME+2	2449
77	AE		02	90	0002B	MOVB	#2, PATHNAME+3	
		74	AE	B4	0002F	CLRW	PATHNAME	
		78	AE	D4	00032	CLRL	PATHNAME+4	
7E	AE		0E	90	00035	MOVB	#14, SHORT_PATH+2	
7F	AE		02	90	00039	MOVB	#2, SHORT_PATH+3	
		7C	AE	B4	0003D	CLRW	SHORT_PATH	
		FC	AD	D4	00040	CLRL	SHORT_PATH+4	
7E	AE		0E	90	00043	MOVB	#14, SHORT_PATH+2	
7F	AE		02	90	00047	MOVB	#2, SHORT_PATH+3	
		7C	AE	B4	0004B	CLRW	SHORT_PATH	
		FC	AD	D4	0004E	CLRL	SHORT_PATH+4	
20	AE	7C	AE	9E	00051	MOVAB	SHORT_PATH, ATTR_LIST+20	
76	AE		0E	90	00056	MOVB	#14, PATHNAME+2	
77	AE		02	90	0005A	MOVB	#2, PATHNAME+3	
		74	AE	B4	0005E	CLRW	PATHNAME	
		78	AE	D4	00061	CLRL	PATHNAME+4	
34	AE	74	AE	9E	00064	MOVAB	PATHNAME, ATTR_LIST+40	
48	AE		6E	9E	00069	MOVAB	ROOT, ATTR_LIST+60	
5C	AE	04	AE	9E	0006D	MOVAB	FORMAT, ATTR_LIST+80	
70	AE	08	AE	9E	00072	MOVAB	LITERALS, ATTR_LIST+100	
		0C	AE	9F	00077	PUSHAB	ATTR_LIST	2457
		04	AC	9F	0007A	PUSHAB	RECORD LOC	
		0144	CB	9F	0007D	PUSHAB	324(CTX)	
00000000G	00		03	FB	00081	CALLS	#3, CDD\$GET_ATT	
	0A		50	E8	00088	BLBS	STATUS, 1\$	2458
		001C823C	50	DD	0008B	PUSHL	STATUS	2460
			8F	DD	0008D	PUSHL	#1868348	
			2C	11	00093	BRB	5\$	
	05	18	AE	E8	00095	BLBS	SHORT_PATH_STATUS, 2\$	2461
		18	AE	DD	00099	PUSHL	SHORT_PATH_STATUS	2463
			1D	11	0009C	BRB	4\$	
0140	CB	7C	AE	9E	0009E	MOVAB	SHORT_PATH, 320(CTX)	2464
	10	2C	AE	E9	000A4	BLBC	PATHNAME_STATUS, 3\$	2465
	0C	54	AE	E9	000A8	BLBC	FORMAT_STATUS, 3\$	2469
	01	04	AE	D1	000AC	CMPL	FORMAT, #1	2470
			13	13	000B0	BEQL	6\$	
	03	04	AE	D1	000B2	CMPL	FORMAT, #3	2471
			0D	13	000B6	BEQL	6\$	
		2C	AE	DD	000B8	PUSHL	PATHNAME_STATUS	2473
		001C8234	8F	DD	000BB	PUSHL	#1868340	
	66		02	FB	000C1	CALLS	#2, CRE_DIAG	
			04	000C4	RET			
00000000'	EF	68	AE	D1	000C5	CMPL	LITERALS_STATUS, CDD__ATTNOTFND	2474
			0A	13	000CD	BEQL	7\$	
		001C824C	8F	DD	000CF	PUSHL	#1868364	2476
	66		01	FB	000D5	CALLS	#1, CRE_DIAG	
			04	000D8	RET			
	0E	40	AE	E9	000D9	BLBC	ROOT_STATUS, 8\$	2480
			7E	7C	000DD	CLRQ	-(SPT)	2482
			01	DD	000DF	PUSHL	#1	
		0C	AE	DD	000E1	PUSHL	ROOT	
0160	C6		04	FB	000E4	CALLS	#4, CRE_FIELD	
			09	11	000E9	BRB	9\$	
		001C821A	8F	DD	000EB	PUSHL	#1868314	2484
	66		01	FB	000F1	CALLS	#1, CRE_DIAG	



CRE\_TRANS  
V04=000

User Literal Definitions

L 3  
16-Sep-1984 01:08:56  
14-Sep-1984 13:10:31

VAX-11 Bliss-32 V4.0-742  
[SORT32.SRC]CRETRANS.B32;1

Page 30  
(10)

67	74	AE 9F 000F4	9\$:	PUSHAB	PATHNAME	:	2486
		01 FB 000F7		CALLS	#1, STR\$FREE1_DX	:	
67	7C	AE 9F 000FA		PUSHAB	SHORT_PATH	:	2487
		01 FB 000FD		CALLS	#1, STR\$FREE1_DX	:	
		04 00100		RET		:	2488
50		0000 00101	10\$:	.WORD	Save nothing	:	2445
50	08	AC D0 00103		MOVL	8(AP), R0	:	
	04	A0 D0 00107		MOVL	4(R0), R0	:	
	F8	A0 9F 0010B		PUSHAB	SHORT_PATH	:	
	F0	A0 9F 0010E		PUSHAB	PATHNAME	:	
		02 DD 00111		PUSHL	#2	:	
		5E DD 00113		PUSHL	SP	:	
F89C	7E	04 AC 7D 00115		MOVQ	4(AP), -(SP)	:	
	CF	03 FB 00119		CALLS	#3, FREE_DYN	:	
		04 0011E		RET		:	

; Routine Size: 287 bytes, Routine Base: SOR\$R0\_CODE + 06A8



```

897 2489 1 ROUTINE CRE_HISTORY
898 2490 1 (
899 2491 1   ENTITY: REF VECTOR[2]
900 2492 1   ): CAL_CTXREG NOVALUE =
901 2493 1 ++
902 2494 1
903 2495 1 FUNCTIONAL DESCRIPTION:
904 2496 1
905 2497 1   This routine creates a history entry for the CDD.
906 2498 1
907 2499 1 FORMAL PARAMETERS:
908 2500 1
909 2501 1   ENTITY -      Entity parameter describing the item
910 2502 1
911 2503 1 IMPLICIT INPUTS:
912 2504 1
913 2505 1   NONE
914 2506 1
915 2507 1 IMPLICIT OUTPUTS:
916 2508 1
917 2509 1   NONE
918 2510 1
919 2511 1 ROUTINE VALUE:
920 2512 1
921 2513 1   Status from call to CDD$CREATE_HISTORY.
922 2514 1
923 2515 1 SIDE EFFECTS:
924 2516 1
925 2517 1   NONE
926 2518 1
927 2519 1 --
928 2520 2 BEGIN
929 2521 2 EXTERNAL REGISTER
930 2522 2   CTX = COM_REG_CTX: REF CTX_BLOCK;
931 2523 2 EXTERNAL ROUTINE
932 2524 2   SOR$STAT;
933 2525 2 LOCAL
934 2526 2   VERSION:      VECTOR[2],      ! Descriptor for version
935 2527 2   BUFFER:        VECTOR[32,BYTE],
936 2528 2   READING:       VECTOR[2]
937 2529 2   INITIAL (%CHARCOUNT('Reading'), UPLIT BYTE('Reading')),
938 2530 2   STATUS;
939 2531 2 MACRO
940 2532 2   FAC = 'VAX-11 SORT/MERGE ' %;
941 2533 2
942 2534 2   VERSION[0] = 0;
943 2535 2   STATUS = SOR$STAT(%REF(STAT_K_IDENT), VERSION[1], %REF(CTX[BASE_]));
944 2536 2   IF .STATUS THEN VERSION[0] = CH$RCHAR A(VERSION[1]);
945 2537 2   VERSION[0] = MINU( .VERSION[0], %ALLOCATION(BUFFER) - %CHARCOUNT(FAC) );
946 2538 2   CH$MOVE(%CHARCOUNT(FAC), UPLIT BYTE(FAC), BUFFER[0]);
947 2539 2   CH$MOVE(.VERSION[0], .VERSION[1], BUFFER[0] + %CHARCOUNT(FAC));
948 2540 2   VERSION[0] = .VERSION[0] + %CHARCOUNT(FAC);
949 2541 2   VERSION[1] = BUFFER[0];
950 2542 2
951 P 2543 2   STATUS = $CDD$CREATE_HISTORY (context = CTX[COM_CDD_CTX],
952 P 2544 2     entity = ENTITY[0],
953 P 2545 2     facility = VERSION[0],
```



```
: 954      P 2546 2      access = READING,  
: 955      P 2547 2      program = 0,  
: 956      2548 2      description = 0);  
: 957      2549 2  
: 958      2550 2      ! Ignore the returned status.  
: 959      2551 2  
: 960      2552 2      !IF NOT .STATUS THEN CRE_DIAG(SOR$_CDDERROR, .STATUS);  
: 961      2553 2  
: 962      2554 1      END;
```

```
52 45 4D 2F 54 52 4F 53 67 6E 69 64 61 65 52 007C7 P.AAD: .ASCII \Reading\  
20 58 41 56 007CE P.AAE: .ASCII \VAX-11 SORT/MERGE \  
20 45 47 007DD
```

.EXTRN SOR\$STAT, CDD\$CREATE\_HISTORY

007C 00000 CRE\_HISTORY:

		5E	38	C2	00002	.WORD	Save R2,R3,R4,R5,R6	: 2489
	08	AE	07	D0	00005	SUBL2	#56, SP	: 2520
	0C	AE	DB	AF	9E 00009	MOVL	#7, READING	: 2534
			30	AE	D4 0000E	MOVAB	P.AAD, READING+4	: 2535
	04	AE	5B	D0	00011	CLRL	VERSION	
			04	AE	9F 00015	MOVL	CTX, 4(SP)	
			38	AE	9F 00018	PUSHAB	4(SP)	
			08	AE	D4 0001B	PUSHAB	VERSION+4	
			08	AE	9F 0001E	CLRL	8(SP)	
				AE	9F 0001E	PUSHAB	8(SP)	
00000000G		00	03	FB	00021	CALLS	#3, SOR\$STAT	
		56	50	D0	00028	MOVL	R0, STATUS	
		08	56	E9	0002B	BLBC	STATUS, 1\$	: 2536
	30	AE	34	BE	9A 0002E	MOVZBL	@VERSION+4, VERSION	
			34	AE	D6 00033	INCL	VERSION+4	
		50	30	AE	D0 00036	MOVL	VERSION, R0	: 2537
		0E		50	D1 0003A	CMPL	R0, #14	
				03	1B 0003D	BLEQU	2\$	
		50	0E	D0	0003F	MOVL	#14, R0	
	30	AE	50	D0	00042	MOVL	R0, VERSION	
10	AE	A4	12	28	00046	MOVC3	#18, P.AAE, BUFFER	: 2538
22	AE	34	AE	28	0004C	MOVC3	VERSION, @VERSION+4, BUFFER+18	: 2539
		30	12	C0	00053	ADDL2	#18, VERSION	: 2540
		34	AE	9E	00057	MOVAB	BUFFER, VERSION+4	: 2541
			7E	7C	0005C	CLRQ	-(SP)	: 2548
			10	AE	9F 0005E	PUSHAB	READING	
			3C	AE	9F 00061	PUSHAB	VERSION	
			04	AC	DD 00064	PUSHL	ENTITY	
			0144	CB	9F 00067	PUSHAB	324(CTX)	
00000000G		00	06	FB	0006B	CALLS	#6, CDD\$CREATE_HISTORY	
		56	50	D0	00072	MOVL	R0, STATUS	
				04	00075	RET		: 2554

; Routine Size: 118 bytes, Routine Base: SOR\$R0\_CODE + 07E0



```
User Literal Definitions

: 964      2555 1 GLOBAL ROUTINE CRE_EXTRACT_RECORD
: 965      2556 1 (
: 966      2557 1     PATH: REF VECTOR[2],
: 967      2558 1     PRM0,           ! Parameter to SOR$$DEF_FIELD
: 968      2559 1     PRM1           ! Parameter to SOR$$DEF_FIELD
: 969      2560 1 ): CAL_CTXREG =
: 970      2561 1 ++
: 971      2562 1
: 972      2563 1 FUNCTIONAL DESCRIPTION:
: 973      2564 1
: 974      2565 1     This routine looks up a record description in the CDD.
: 975      2566 1
: 976      2567 1 FORMAL PARAMETERS:
: 977      2568 1
: 978      2569 1     PATH           - descriptor for the CDD pathname of the target record
: 979      2570 1
: 980      2571 1 IMPLICIT INPUTS:
: 981      2572 1
: 982      2573 1     NONE
: 983      2574 1
: 984      2575 1 IMPLICIT OUTPUTS:
: 985      2576 1
: 986      2577 1     NONE
: 987      2578 1
: 988      2579 1 ROUTINE VALUE:
: 989      2580 1
: 990      2581 1     Address of the pathname description block, or zero if errors occurred.
: 991      2582 1
: 992      2583 1 SIDE EFFECTS:
: 993      2584 1
: 994      2585 1     This routine will access the CDD.
: 995      2586 1
: 996      2587 1 --
: 997      2588 2 BEGIN
: 998      2589 2 EXTERNAL REGISTER
: 999      2590 2     CTX = COM_REG_CTX: REF CTX_BLOCK;
1000      2591 2 GLOBAL REGISTER
1001      2592 2     PRM = REG_PRM: REF VECTOR[2];
1002      2593 2
1003      2594 2 LITERAL
1004      2595 2     ACCEPTABLE_LEVEL = 4;           ! The acceptable CORE_LEVEL
1005      2596 2 BUILTIN
1006      2597 2     ACTUALCOUNT,
1007      2598 2     NULLPARAMETER;
1008      2599 2 LOCAL
1009      2600 2     CORE_LEVEL,
1010      2601 2     PROTOCOL: VECTOR[2],
1011      2602 2     BUFFER: VECTOR[32,BYTE],
1012      2603 2     RECORD_BASE,
1013      2604 2     STATUS;
1014      2605 2
1015      2606 2 STATUS = $CDD$SIGN IN(context = CTX[COM CDD CTX]);
1016      2607 2 IF NOT .STATUS THEN RETURN CRE_DIAG(SOR$_CDDERROR, .STATUS);
1017      2608 2
1018      2609 2 ! Perform initialization
1019      2610 2
1020      2611 2 PROTOCOL[0] = 32;
```



```

: 1021      2612 2  PROTOCOL[1] = BUFFER;
: 1022      2613 2  CTX[COM_CDD_PATH] = PATH[0];
: 1023      2614 2  IF CDD$_ATTNOTFND EQL 0 THEN CDD$_ATTNOTFND();
: 1024      2615 2
: 1025      2616 2  ! Find the target record description.
: 1026      2617 2
: 1027      P 2618 2  STATUS = $CDD$FIND_NODE(context = CTX[COM_CDD_CTX],
: 1028      P 2619 2  path = PATH[0],
: 1029      P 2620 2  location = RECORD_BASE,
: 1030      2621 2  protocol = PROTOCOL);
: 1031      2622 2  PROTOCOL[0] = 1 +
: 1032      2623 2  (DECR I FROM 31 TO 0 DO IF .BUFFER[I] NEQ %C' ' THEN EXITLOOP .I);
: 1033      2624 2
: 1034      2625 2  IF NOT .STATUS THEN RETURN CRE_DIAG(SOR$_CDDERROR, .STATUS);
: 1035      2626 2
: 1036      2627 2  ! Validate that this is a record description and that we should
: 1037      2628 2  be able to read it.
: 1038      2629 2
: 1039      2630 2  IF NOT CH$EQL(%CHARCOUNT('CDD$RECORD'), UPLIT BYTE('CDD$RECORD'),
: 1040      2631 2  .PROTOCOL[0], .PROTOCOL[1], ' ')
: 1041      2632 2  THEN
: 1042      2633 2  RETURN SOR$$ERROR(SOR$_NOTRECORD, 2, PATH[0], PROTOCOL);
: 1043      2634 2
: 1044      P 2635 2  STATUS = $CDD$GET_NUM_ATT(context = CTX[COM_CDD_CTX],
: 1045      P 2636 2  entity = RECORD_BASE,
: 1046      P 2637 2  attribute = CDD$_CORE_LEVEL,
: 1047      2638 2  value = CORE_LEVEL);
: 1048      2639 2  IF NOT .STATUS THEN RETURN CRE_DIAG(SOR$_BADPROTCL, .STATUS);
: 1049      2640 2
: 1050      2641 2  IF .CORE_LEVEL NEQ ACCEPTABLE_LEVEL
: 1051      2642 2  THEN
: 1052      2643 2  RETURN CRE_DIAG(SOR$_UNSUPLEVL);
: 1053      2644 2
: 1054      2645 2  ! Create a history entry
: 1055      2646 2
: 1056      2647 2  CRE_HISTORY(RECORD_BASE);
: 1057      2648 2
: 1058      2649 2  ! Now translate the record
: 1059      2650 2
: 1060      2651 2  PRM = PRM0;
: 1061      2652 2  CRE_TRANS_RECORD(.RECORD_BASE);
: 1062      P 2653 2  STATUS = $CDD$RLSE_LOCKS(context = CTX[COM_CDD_CTX],
: 1063      2654 2  node = RECORD_BASE);
: 1064      2655 2  IF NOT .STATUS THEN RETURN CRE_DIAG(SOR$_CDDERROR, .STATUS);
: 1065      2656 2
: 1066      2657 2  STATUS = $CDD$SIGN_OUT(context = CTX[COM_CDD_CTX]);
: 1067      2658 2  CTX[COM_CDD_CTX] = 0;
: 1068      2659 2  IF NOT .STATUS THEN RETURN CRE_DIAG(SOR$_CDDERROR, .STATUS);
: 1069      2660 2
: 1070      2661 2  RETURN SS$_NORMAL;
: 1071      2662 1  END;
```

44 52 4F 43 45 52 24 44 44 43 00856 P.AAF: .ASCII \CDD\$RECORD\

.EXTRN CDD\$SIGN\_IN, CDD\$FIND\_NODE



				.EXTRN	CDD\$GET_NUM_ATT		
				.EXTRN	CDD\$RLSE_LOCKS, CDD\$SIGN_OUT		
			043C 00000	.ENTRY	CRE_EXTRACT_RECORD, Save R2,R3,R4,R5,R10	: 2555	
	55	F79A	CF 9E 00002	MOVAB	CRE_DIAG, R5	:	
	5E		30 C2 00007	SUBL2	#48, SP	:	
		0144	CB 9F 0000A	PUSHAB	324(CTX)	: 2606	
00000000G	00		01 FB 0000E	CALLS	#1, CDD\$SIGN_IN	:	
	54		50 D0 00015	MOVL	R0, STATUS	:	
	4A		54 E9 00018	BLBC	STATUS, 4\$	: 2607	
28	AE		20 D0 0001B	MOVL	#32, PROTOCOL	: 2611	
2C	AE	08	AE 9E 0001F	MOVAB	BUFFER, PROTOCOL+4	: 2612	
0140	CB	04	AC D0 00024	MOVL	PATH, 320(CTX)	: 2613	
		00000000	EF D5 0002A	TSTL	CDD_--ATTNOTFND	: 2614	
			05 12 00030	BNEQ	1\$	:	
0000V	CF		00 FB 00032	CALLS	#0, CDD ATTNOTFND	:	
		28	AE 9F 00037	PUSHAB	PROTOCOL	: 2621	
		08	AE 9F 0003A	PUSHAB	RECORD_BASE	:	
			7E D4 0003D	CLRL	-(SP)	:	
		04	AC DD 0003F	PUSHL	PATH	:	
		0144	CB 9F 00042	PUSHAB	324(CTX)	:	
00000000G	00		05 FB 00046	CALLS	#5, CDD\$FIND_NODE	:	
	54		50 D0 0004D	MOVL	R0, STATUS	:	
	50		1F D0 00050	MOVL	#31, I	: 2623	
	20	08	AE 40 91 00053	CMPB	BUFFER[I], #32	:	
			06 12 00058	BNEQ	3\$	:	
	F6		50 F4 0005A	SOBGEQ	I, 2\$	:	
	50		01 CE 0005D	MNEGL	#1, R0	:	
28	AE	01	A0 9E 00060	MOVAB	1(R0), PROTOCOL	: 2622	
	7D		54 E9 00065	BLBC	STATUS, 8\$	: 2625	
28	AE		0A 2D 00068	CMPC5	#10, P.AAF, #32, PROTOCOL, @PROTOCOL+4	: 2630	
		2C	BE 0006F			:	
			16 13 00071	BEQL	5\$	:	
		28	AE 9F 00073	PUSHAB	PROTOCOL	: 2633	
		04	AC DD 00076	PUSHL	PATH	:	
			02 DD 00079	PUSHL	#2	:	
		001C8274	8F DD 0007B	PUSHL	#1868404	:	
00000000G	00		04 FB 00081	CALLS	#4, SOR\$ERROR	:	
			04 04 00088	RET		:	
			5E DD 00089	PUSHL	SP	: 2638	
		10000001	8F DD 0008B	PUSHL	#268435457	:	
		0C	AE 9F 00091	PUSHAB	RECORD_BASE	:	
		0144	CB 9F 00094	PUSHAB	324(CTX)	:	
00000000G	00		04 FB 00098	CALLS	#4, CDD\$GET_NUM_ATT	:	
	54		50 D0 0009F	MOVL	R0, STATUS	:	
	0A		54 E8 000A2	BLBS	STATUS, 6\$	: 2639	
			54 DD 000A5	PUSHL	STATUS	:	
		001C8234	8F DD 000A7	PUSHL	#1868340	:	
			56 11 000AD	BRB	10\$	:	
	04		6E D1 000AF	CMPL	CORE_LEVEL, #4	: 2641	
			0A 13 000B2	BEQL	7\$	:	
		001C8284	8F DD 000B4	PUSHL	#1868420	: 2643	
	65		01 FB 000BA	CALLS	#1, CRE_DIAG	:	
			04 04 000BD	RET		:	
		04	AE 9F 000BE	PUSHAB	RECORD_BASE	: 2647	
07E0	C5		01 FB 000C1	CALLS	#1, CRE_HISTORY	:	
	5A	08	AC 9E 000C6	MOVAB	PRM0, PRM	: 2651	



CRE\_TRANS  
V04=000

User Literal Definitions

E 4  
16-Sep-1984 01:08:56  
14-Sep-1984 13:10:31

VAX-11 Bliss-32 V4.0-742  
[SORT32.SRC]CRETRANS.B32;1

Page 36  
(12)

06A8	C5	04	AE	DD	000CA	PUSHL	RECORD_BASE	:	2652
		01	FB	000CD		CALLS	#1, CRE_TRANS_RECORD	:	
		04	AE	9F	000D2	PUSHAB	RECORD_BASE	:	2654
		7E	D4	000D5	CLRL	-(SP)		:	
00000000G	00	0144	CB	9F	000D7	PUSHAB	324(CTX)	:	
	54		03	FB	000DB	CALLS	#3, CDD\$RLSE_LOCKS	:	
	15		50	D0	000E2	MOVL	R0, STATUS	:	
			54	E9	000E5	BLBC	STATUS, 9\$	:	2655
00000000G	00	0144	CB	9F	000E8	PUSHAB	324(CTX)	:	2657
	54		01	FB	000EC	CALLS	#1, CDD\$SIGN_OUT	:	
			50	D0	000F3	MOVL	R0, STATUS	:	
		0144	CB	D4	000F6	CLRL	324(CTX)	:	2658
	0C		54	E8	000FA	BLBS	STATUS, 11\$	:	2659
			54	DD	000FD	PUSHL	STATUS	:	
	001C823C		8F	DD	000FF	PUSHL	#1868348	:	
	65		02	FB	00105	CALLS	#2, CRE_DIAG	:	
				04	00108	RET		:	
	50		01	D0	00109	MOVL	#1, R0	:	2661
				04	0010C	RET		:	2662

; Routine Size: 269 bytes,      Routine Base: SOR\$RO\_CODE + 0860



```

: 1073      2663 1 ROUTINE CRE_CLEAN_UP: CAL_CTXREG NOVALUE =
: 1074      2664 1
: 1075      2665 1 ++
: 1076      2666 1
: 1077      2667 1 FUNCTIONAL DESCRIPTION:
: 1078      2668 1
: 1079      2669 1 Release resources allocated by this module.
: 1080      2670 1
: 1081      2671 1 FORMAL PARAMETERS:
: 1082      2672 1
: 1083      2673 1 NONE
: 1084      2674 1
: 1085      2675 1 IMPLICIT INPUTS:
: 1086      2676 1
: 1087      2677 1 NONE
: 1088      2678 1
: 1089      2679 1 IMPLICIT OUTPUTS:
: 1090      2680 1
: 1091      2681 1 NONE
: 1092      2682 1
: 1093      2683 1 ROUTINE VALUE:
: 1094      2684 1
: 1095      2685 1 NONE
: 1096      2686 1
: 1097      2687 1 SIDE EFFECTS:
: 1098      2688 1
: 1099      2689 1 NONE
: 1100      2690 1
: 1101      2691 1 --
: 1102      2692 2 BEGIN
: 1103      2693 2 EXTERNAL REGISTER
: 1104      2694 2 CTX = COM_REG_CTX: REF CTX_BLOCK;
: 1105      2695 2 LOCAL
: 1106      2696 2 STATUS;
: 1107      2697 2
: 1108      2698 2 IF .CTX[COM_CDD_CTX] NEQ 0
: 1109      2699 2 THEN
: 1110      2700 3 BEGIN
: 1111      2701 3 STATUS = $CDD$SIGN_OUT(context = CTX[COM_CDD_CTX]);
: 1112      2702 3 CTX[COM_CDD_CTX] = 0;
: 1113      2703 3 IF NOT .STATUS THEN CRE_DIAG(SOR$_CDDERROR, .STATUS);
: 1114      2704 2 END;
: 1115      2705 2
: 1116      2706 1 END;
```

		0004 00000	CRE_CLEAN_UP:			
				.WORD	Save R2	: 2663
52	0144	CB 9E 00002	MOVAB	324(CTX), R2		: 2698
		62 D5 00007	TSTL	(R2)		
		1B 13 00009	BEQL	1\$		
		52 DD 0000B	PUSHL	R2		: 2701
00000000G	00	01 FB 0000D	CALLS	#1, CDD\$SIGN_OUT		
		62 D4 00014	CLRL	(R2)		: 2702



	OD	50	E8	00016	BLBS	STATUS, 1\$
		50	DD	00019	PUSHL	STATUS
	001C823C	8F	DD	0001B	PUSHL	#1868348
F66D	CF	02	FB	00021	CALLS	#2, CRE_DIAG
			04	00026	1\$: RET	

: 2703  
:  
:  
:  
:  
:  
:  
: 2706

```
; Routine Size: 39 bytes,      Routine Base: SOR$RO_CODE + 096D
```



## User Literal Definitions

```
: 1118      2707 1 EXTERNAL ROUTINE
: 1119      2708 1   LIB$FIND_IMAGE_SYMBOL:   ADDRESSING_MODE(GENERAL);
: 1120      2709 1 BIND
: 1121      2710 1   CDDSHR1 = UPLIT BYTE('CDDSHR');
: 1122      2711 1 MACRO
: 1123      2712 1   CDDSHR_DECL = VECTOR[2] INITIAL (%CHARCOUNT('CDDSHR'), CDDSHR1) %;
: 1124      2713 1 MACRO
: 1125      2714 1   NAMSTR_DECL(X) = VECTOR[2] INITIAL (%CHARCOUNT(X), UPLIT BYTE(X)) %;
: 1126      2715 1 MACRO
: 1127      2716 1   CDD_(X) =
: 1128      2717 1   BEGIN
: 1129      2718 1   OWN Z: INITIAL(0);
: 1130      2719 1   LOCAL STATUS;
: 1131      2720 1   BUILTIN AP, CALLG;
: 1132      2721 1   IF .Z EQL 0
: 1133      2722 1   THEN
: 1134      2723 1   BEGIN
: 1135      2724 1   LOCAL CDDSHR: CDDSHR_DECL;
: 1136      2725 1   LOCAL NAMSTR: NAMSTR_DECL(X);
: 1137      2726 1   STATUS = LIB$FIND_IMAGE_SYMBOL(CDDSHR, NAMSTR, Z);
: 1138      2727 1   IF NOT .STATUS THEN RETURN SOR$ERROR(SOR$_SHR_SYSEERROR,0,.STATUS);
: 1139      2728 1   END;
: 1140      2729 1   RETURN CALLG(.AP, .Z);
: 1141      2730 1   END %;
: 1142      2731 1 GLOBAL ROUTINE CDD$CREATE_HISTORY = CDD_('CDD$CREATE_HISTORY');
```

```
                                .PSECT SOR$RW_PICDATA,NOEXE, PIC,2
                                00000000 00004 Z: .LONG 0
                                .PSECT SOR$RO_CODE,NOWRT, SHR, PIC,2
54 53 49 48 5F 45 54 41 45 52 48 53 44 44 43 00994 P.AAG: .ASCII \CDDSHR\
: 52 43 24 44 44 43 0099A P.AAH: .ASCII \CDD$CREATE_HISTORY\
: 59 52 4F 009A9
                                CDDSHR1=
                                .EXTRN P.AAG LIB$FIND_IMAGE_SYMBOL
                                .ENTRY CDD$CREATE_HISTORY, Save R2
                                52 00000000' EF 9E 00002 MOVAB Z, R2
                                5E 10 C2 00009 SUBL2 #16, SP
                                62 D5 0000C TSTL Z
                                35 12 0000E BNEQ 1$
                                08 AE 06 D0 00010 MOVL #6, CDDSHR
                                0C AE D1 AF 9E 00014 MOVAB CDDSHR1, CDDSHR+4
                                6E 12 D0 00019 MOVL #18, NAMSTR
                                04 AE CF AF 9E 0001C MOVAB P.AAH, NAMSTR+4
                                52 DD 00021 PUSHL R2
                                04 AE 9F 00023 PUSHAB NAMSTR
                                10 AE 9F 00026 PUSHAB CDDSHR
                                00000000G 00 03 FB 00029 CALLS #3, LIB$FIND_IMAGE_SYMBOL
                                12 50 E8 00030 BLBS STATUS, 1$
                                50 DD 00033 PUSHL STATUS
                                7E D4 00035 CLRL -(SP)
```



```

00000000G 00 001C11B4 8F DD 00037 PUSHL #1839540
03 FB 0003D CALLS #3, SOR$$ERROR
04 00044 RET
00 B2 6C FA 00045 1$: CALLG (AP), @Z
04 00049 RET

```

; Routine Size: 74 bytes, Routine Base: SOR\$RO\_CODE + 09AC

```
; 1143      2732  1 GLOBAL ROUTINE CDD$FIND_NODE      = CDD_('CDD$FIND_NODE');
```

```
.PSECT SORSRW_PICDATA,NOEXE, PIC,2
```

```
00000000 00008 Z:      .LONG  0
```

```

.PSECT SORSRO_CODE,NOWRT, SHR, PIC,2

```

```
45 44 4F 4E 5F 44 4E 49 46 24 44 44 43 009F6 P.AAI: .ASCII \CDD$FIND_NODE\
```

			0004 00000	.ENTRY	CDD\$FIND_NODE, Save R2
	52	00000000'	EF 9E 00002	MOVAB	Z, R2
	5E		10 C2 00009	SUBL2	#16, SP
			62 D5 0000C	TSTL	Z
			36 12 0000E	BNEQ	1\$
08	AE		06 D0 00010	MOVL	#6, CDDSHR
0C	AE	FF79	CF 9E 00014	MOVAB	CDDSHR1, CDDSHR+4
	6E		0D D0 0001A	MOVL	#13, NAMSTR
04	AE	D3	AF 9E 0001D	MOVAB	P.AAI, NAMSTR+4
			52 DD 00022	PUSHL	R2
		04	AE 9F 00024	PUSHAB	NAMSTR
		10	AE 9F 00027	PUSHAB	CDDSHR
00000000G	00		03 FB 0002A	CALLS	#3, LIB\$FIND_IMAGE_SYMBOL
	12		50 E8 00031	BLBS	STATUS, 1\$
			50 DD 00034	PUSHL	STATUS
			7E D4 00036	CLRL	-(SP)
		001C11B4	8F DD 00038	PUSHL	#1839540
00000000G	00		03 FB 0003E	CALLS	#3, SOR\$\$ERROR
			04 00045	RET	
00	B2		6C FA 00046 1\$:	CALLG	(AP), @Z
			04 0004A	RET	

; Routine Size: 75 bytes, Routine Base: SOR\$RC\_CODE + 0A03

```
; 1144      2733  1 GLOBAL ROUTINE CDD$GET_ATT$ = CDD_('CDD$GET_ATT$');
```

```
.PSECT SORSRW_PICDATA,NOEXE, PIC,2
```

```
00000000 0000c Z:      .LONG  0
```

```

.PSECT SORSRO_CODE,NOWRT, SHR, PIC,2

```



53 54 54 41 5F 54 45 47 24 44 44 43 00A4E P.AAJ: .ASCII \CDD\$GET\_ATT\$

				0004 00000	.ENTRY	CDD\$GET_ATT\$, Save R2		2733
	52	00000000'	EF	9E 00002	MOVAB	Z, R2		
	5E		10	C2 00009	SUBL2	#16, SP		
			62	D5 0000C	TSTL	Z		
			36	12 0000E	BNEQ	1\$		
08	AE		06	D0 00010	MOVL	#6, CDDSHR		
0C	AE	FF22	CF	9E 00014	MOVAB	CDDSHR1, CDDSHR+4		
	6E		0C	D0 0001A	MOVL	#12, NAMSTR		
04	AE	D4	AF	9E 0001D	MOVAB	P.AAJ, NAMSTR+4		
			52	DD 00022	PUSHL	R2		
		04	AE	9F 00024	PUSHAB	NAMSTR		
		10	AE	9F 00027	PUSHAB	CDDSHR		
00000000G	00		03	FB 0002A	CALLS	#3, LIB\$FIND_IMAGE_SYMBOL		
	12		50	E8 00031	BLBS	STATUS, 1\$		
			50	DD 00034	PUSHL	STATUS		
			7E	D4 00036	CLRL	-(SP)		
00000000G	00	001C11B4	8F	DD 00038	PUSHL	#1839540		
			03	FB 0003E	CALLS	#3, SOR\$ERROR		
			04	00045	RET			
00	B2		6C	FA 00046 1\$:	CALLG	(AP), @Z		
			04	0004A	RET			

; Routine Size: 75 bytes, Routine Base: SOR\$RO\_CODE + 0A5A

; 1145 2734 1 GLOBAL ROUTINE CDD\$GET\_ENTITY\_CELL = CDD\_('CDD\$GET\_ENTITY\_CELL');

					.PSECT	SOR\$RW_PICDATA, NOEXE, PIC, 2		
				00000000 00010 Z:	.LONG	0		
					.PSECT	SOR\$RO_CODE, NOWRT, SHR, PIC, 2		
5F 59 54 49 54 4E 45 5F 54 45 47 24 44 44 43 00AA5 P.AAK:					.ASCII	\CDD\$GET_ENTITY_CELL\		
			4C 4C 45 43 00AB4					

				0004 00000	.ENTRY	CDD\$GET_ENTITY_CELL, Save R2		2734
	52	00000000'	EF	9E 00002	MOVAB	Z, R2		
	5E		10	C2 00009	SUBL2	#16, SP		
			62	D5 0000C	TSTL	Z		
			36	12 0000E	BNEQ	1\$		
08	AE		06	D0 00010	MOVL	#6, CDDSHR		
0C	AE	FEC4	CF	9E 00014	MOVAB	CDDSHR1, CDDSHR+4		
	6E		13	D0 0001A	MOVL	#19, NAMSTR		
04	AE	CD	AF	9E 0001D	MOVAB	P.AAK, NAMSTR+4		
			52	DD 00022	PUSHL	R2		
		04	AE	9F 00024	PUSHAB	NAMSTR		
		10	AE	9F 00027	PUSHAB	CDDSHR		



```
00000000G 00      03 FB 0002A    CALLS #3, LIB$FIND_IMAGE_SYMBOL
                12      50 E8 00031    BLBS STATUS, 1$
                50 DD 00034    PUSHL STATUS
                7E D4 00036    CLRL -(SP)
00000000G 00      8F DD 00038    PUSHL #1839540
                00      03 FB 0003E    CALLS #3, SOR$$ERROR
                00 B2      04 00045    RET
                6C FA 00046 1$:    CALLG (AP), @Z
                04 0004A    RET
```

; Routine Size: 75 bytes, Routine Base: SOR\$RO\_CODE + 0AB8

; 1146 2735 1 GLOBAL ROUTINE CDD\$GET\_ENTITY\_LIST\_ATT = CDD\_('CDD\$GET\_ENTITY\_LIST\_ATT');

```
                                .PSECT SOR$RW_PICDATA,NOEXE, PIC,2
                                00000000 00014 Z: .LONG 0
                                .PSECT SOR$RO_CODE,NOWRT, SHR, PIC,2
5F 59 54 49 54 4E 45 5F 54 45 47 24 44 44 43 00B03 P.AAL: .ASCII \CDD$GET_ENTITY_LIST_ATT\
54 54 41 5F 54 53 49 4C 00B12
```

```
                                .ENTRY CDD$GET_ENTITY_LIST_ATT, Save R2
                                0004 00000    MOVAB Z, R2
                                52 00000000' EF 9E 00002    SUBL2 #16, SP
                                5E      10 C2 00009    TSTL Z
                                62 D5 0000C    BNEQ 1$
                                36 12 0000E    MOVL #6, CDDSHR
                                08 AE      06 D0 00010    MOVAB CDDSHR1, CDDSHR+4
                                0C AE FE62 CF 9E 00014    MOVL #23, NAMSTR
                                6E      17 D0 0001A    MOVAB P.AAL, NAMSTR+4
                                04 AE C9 AF 9E 0001D    PUSHL R2
                                04      52 DD 00022    PUSHAB NAMSTR
                                10      AE 9F 00024    PUSHAB CDDSHR
                                00000000G 00      03 FB 0002A    CALLS #3, LIB$FIND_IMAGE_SYMBOL
                                12      50 E8 00031    BLBS STATUS, 1$
                                50 DD 00034    PUSHL STATUS
                                7E D4 00036    CLRL -(SP)
                                00000000G 00      8F DD 00038    PUSHL #1839540
                                00      03 FB 0003E    CALLS #3, SOR$$ERROR
                                00 B2      04 00045    RET
                                6C FA 00046 1$:    CALLG (AP), @Z
                                04 0004A    RET
```

; Routine Size: 75 bytes, Routine Base: SOR\$RO\_CODE + 0B1A

; 1147 2736 1 GLOBAL ROUTINE CDD\$GET\_NUM\_ATT = CDD\_('CDD\$GET\_NUM\_ATT');



```

                                .PSECT SOR$RW_PICDATA,NOEXE, PIC,2
                                00000000 00018 Z: .LONG 0
                                .PSECT SOR$RO_CODE,NOWRT, SHR, PIC,2
54 54 41 5F 4D 55 4E 5F 54 45 47 24 44 44 43 00B65 P.AAM: .ASCII \CDD$GET_NUM_ATT\
                                .ENTRY CDD$GET_NUM_ATT, Save R2
                                MOVAB Z, R2
                                SUBL2 #16, SP
                                TSTL Z
                                BNEQ 1$
                                MOVL #6, CDDSHR
                                MOVAB CDDSHR1, CDDSHR+4
                                MOVL #15, NAMSTR
                                MOVAB P.AAM, NAMSTR+4
                                PUSHL R2
                                PUSHAB NAMSTR
                                PUSHAB CDDSHR
                                CALLS #3, LIB$FIND_IMAGE_SYMBOL
                                BLBS STATUS, 1$
                                PUSHL STATUS
                                CLRL -(SP)
                                PUSHL #1839540
                                CALLS #3, SOR$$ERROR
                                RET
                                00000000' EF 9E 00002 .ENTRY CDD$GET_STRING_ATT, Save R2
                                5E 10 C2 00009 MOVAB Z, R2
                                62 D5 0000C SUBL2 #16, SP
                                36 12 0000E TSTL Z
                                08 AE FE08 06 D0 00010 BNEQ 1$
                                0C AE 0F D0 00014 MOVL #6, CDDSHR
                                6E 0F D0 0001A MOVAB CDDSHR1, CDDSHR+4
                                04 AE D1 AF 9E 0001D MOVL #15, NAMSTR
                                52 DD 00022 MOVAB P.AAM, NAMSTR+4
                                04 AE 9F 00024 PUSHL R2
                                10 AE 9F 00027 PUSHAB NAMSTR
                                00000000G 00 03 FB 0002A PUSHAB CDDSHR
                                12 50 E8 00031 CALLS #3, LIB$FIND_IMAGE_SYMBOL
                                50 DD 00034 BLBS STATUS, 1$
                                7E D4 00036 PUSHL STATUS
                                001C11B4 8F DD 00038 CLRL -(SP)
                                00000000G 00 03 FB 0003E PUSHL #1839540
                                00 B2 6C FA 00046 CALLS #3, SOR$$ERROR
                                1$: 04 00045 RET
                                04 0004A RET

```

; Routine Size: 75 bytes, Routine Base: SOR\$RO\_CODE + 0B74

; 1148 2737 1 GLOBAL ROUTINE CDD\$GET\_STRING\_ATT = CDD\_('CDD\$GET\_STRING\_ATT');

```

                                .PSECT SOR$RW_PICDATA,NOEXE, PIC,2
                                00000000 0001C Z: .LONG 0
                                .PSECT SOR$RO_CODE,NOWRT, SHR, PIC,2
5F 47 4E 49 52 54 53 5F 54 45 47 24 44 44 43 00BBF P.AAN: .ASCII \CDD$GET_STRING_ATT\
                                54 54 41 00BCE
                                .ENTRY CDD$GET_STRING_ATT, Save R2
                                MOVAB Z, R2
                                SUBL2 #16, SP
                                TSTL Z
                                BNEQ 1$
                                MOVL #6, CDDSHR
                                MOVAB CDDSHR1, CDDSHR+4
                                52 00000000' EF 9E 00002
                                5E 10 C2 00009
                                62 D5 0000C
                                36 12 0000E
                                08 AE FDAB 06 D0 00010
                                0C AE CF 9E 00014

```



04	6E	CE	12	DO	0001A	MOVL	#18, NAMSTR	:
	AE		AF	9E	0001D	MOVAB	P.AAN, NAMSTR+4	:
			52	DD	00022	PUSHL	R2	:
		04	AE	9F	00024	PUSHAB	NAMSTR	:
		10	AE	9F	00027	PUSHAB	CDDSHR	:
00000000G	00		03	FB	0002A	CALLS	#3, LIB\$FIND_IMAGE_SYMBOL	:
	12		50	E8	00031	BLBS	STATUS, 1\$	:
			50	DD	00034	PUSHL	STATUS	:
			7E	D4	00036	CLRL	-(SP)	:
00000000G	00	001C11B4	8F	DD	00038	PUSHL	#1839540	:
			03	FB	0003E	CALLS	#3, SOR\$\$ERROR	:
				04	00045	RET		:
00	B2		6C	FA	00046 1\$:	CALLG	(AP), @Z	:
			04	0004A		RET		:

; Routine Size: 75 bytes, Routine Base: SOR\$RO\_CODE + 0BD1

; 1149 2738 1 GLOBAL ROUTINE CDD\$RLSE\_LOCKS = CDD\_('CDD\$RLSE\_LOCKS');

.PSECT SOR\$RW\_PICDATA,NOEXE, PIC,2

00000000 00020 Z:

.LONG 0

.PSECT SOR\$RO\_CODE,NOWRT, SHR, PIC,2

53 4B 43 4F 4C 5F 45 53 4C 52 24 44 44 43 00C1C P.AAO: .ASCII \CDD\$RLSE\_LOCKS\

			0004	00000	.ENTRY	CDD\$RLSE_LOCKS, Save R2	:	
	52	00000000'	EF	9E	00002	MOVAB	Z, R2	:
	5E		10	C2	00009	SUBL2	#16, SP	:
			62	D5	0000C	TSTL	Z	:
			36	12	0000E	BNEQ	1\$	:
08	AE		06	DO	00010	MOVL	#6, CDDSHR	:
0C	AE	FD52	CF	9E	00014	MOVAB	CDDSHR1, CDDSHR+4	:
	6E		0E	DO	0001A	MOVL	#14, NAMSTR	:
04	AE	D2	AF	9E	0001D	MOVAB	P.AAO, NAMSTR+4	:
			52	DD	00022	PUSHL	R2	:
		04	AE	9F	00024	PUSHAB	NAMSTR	:
		10	AE	9F	00027	PUSHAB	CDDSHR	:
00000000G	00		03	FB	0002A	CALLS	#3, LIB\$FIND_IMAGE_SYMBOL	:
	12		50	E8	00031	BLBS	STATUS, 1\$	:
			50	DD	00034	PUSHL	STATUS	:
			7E	D4	00036	CLRL	-(SP)	:
00000000G	00	001C11B4	8F	DD	00038	PUSHL	#1839540	:
			03	FB	0003E	CALLS	#3, SOR\$\$ERROR	:
				04	00045	RET		:
00	B2		6C	FA	00046 1\$:	CALLG	(AP), @Z	:
			04	0004A		RET		:

; Routine Size: 75 bytes, Routine Base: SOR\$RO\_CODE + 0C2A



CRE\_TRANS  
V04=000  
; 1150

User Literal Definitions

2739 1 GLOBAL ROUTINE CDD\$SIGN\_IN

N 4  
16-Sep-1984 01:08:56  
14-Sep-1984 13:10:31  
= CDD\_('CDD\$SIGN\_IN');

VAX-11 Bliss-32 V4.0-742  
[SORT32.SRC]CRETRANS.B32;1

Page 45  
(14)

```
                                .PSECT SOR$RW_PICDATA,NOEXE, PIC,2
                                00000000 00024 Z: .LONG 0
                                .PSECT SOR$RO_CODE,NOWRT, SHR, PIC,2
4E 49 5F 4E 47 49 53 24 44 44 43 00C75 P.AAP: .ASCII \CDD$SIGN_IN\
;
                                .ENTRY CDD$SIGN_IN, Save R2
                                MOVAB Z, R2
                                SUBL2 #16, SP
                                TSTL Z
                                BNEQ 1$
                                MOVL #6, CDDSHR
                                MOVAB CDDSHR1, CDDSHR+4
                                MOVL #11, NAMSTR
                                MOVAB P.AAP, NAMSTR+4
                                PUSHL R2
                                PUSHAB NAMSTR
                                PUSHAB CDDSHR
                                CALLS #3, LIB$FIND_IMAGE_SYMBOL
                                BLBS STATUS, 1$
                                PUSHL STATUS
                                CLRL -(SP)
                                PUSHL #1839540
                                CALLS #3, SOR$$ERROR
                                RET
                                CALLG (AP), @Z
                                RET
                                ; 2739
```

; Routine Size: 75 bytes, Routine Base: SOR\$RO\_CODE + 0C80

; 1151 2740 1 GLOBAL ROUTINE CDD\$SIGN\_OUT = CDD\_('CDD\$SIGN\_OUT');

```
                                .PSECT SOR$RW_PICDATA,NOEXE, PIC,2
                                00000000 00028 Z: .LONG 0
                                .PSECT SOR$RO_CODE,NOWRT, SHR, PIC,2
54 55 4F 5F 4E 47 49 53 24 44 44 43 00CCB P.AAQ: .ASCII \CDD$SIGN_OUT\
;
                                .ENTRY CDD$SIGN_OUT, Save R2
                                MOVAB Z, R2
                                SUBL2 #16, SP
                                TSTL Z
                                ; 2740
```



08	AE		36	12	0000E	BNEQ	1\$	
0C	AE	FCA5	06	D0	00010	MOVL	#6, CDDSHR	
	6E		CF	9E	00014	MOVAB	CDDSHR1, CDDSHR+4	
04	AE	D4	0C	D0	0001A	MOVL	#12, NAMSTR	
			AF	9E	0001D	MOVAB	P.AAQ, NAMSTR+4	
		04	52	DD	00022	PUSHL	R2	
		10	AE	9F	00024	PUSHAB	NAMSTR	
			AE	9F	00027	PUSHAB	CDDSHR	
00000000G	00		03	FB	0002A	CALLS	#3, LIB\$FIND_IMAGE_SYMBOL	
	12		50	E8	00031	BLBS	STATUS, 1\$	
			50	DD	00034	PUSHL	STATUS	
			7E	D4	00036	CLRL	-(SP)	
00000000G	00	001C11B4	8F	DD	00038	PUSHL	#1839540	
			03	FB	0003E	CALLS	#3, SOR\$\$ERROR	
			04	00	00045	RET		
00	B2		6C	FA	00046	CALLG	(AP), @Z	
			04	00	0004A	RET		

; Routine Size: 75 bytes, Routine Base: SOR\$RO\_CODE + 0CD7

```
: 1152      2741 1 GLOBAL ROUTINE CDD_ATTNOTFND: NOVALUE =
: 1153      2742 2 BEGIN
: 1154      2743 2 LOCAL CDDSHR: CDDSHR_DECL;
: 1155      2744 2 LOCAL NAMSTR: NAMSTR_DECL('CDD$_ATTNOTFND');
: 1156      2745 2 LOCAL STATUS;
: 1157      2746 2 STATUS = LIB$FIND IMAGE SYMBOL(CDDSHR, NAMSTR, CDD_ATTNOTFND);
: 1158      2747 2 IF NOT .STATUS THEN RETURN SOR$$ERROR(SOR$_SHR_SYSEERROR,0,.STATUS);
: 1159      2748 1 END;
```

44 4E 46 54 4F 4E 54 54 41 5F 24 44 44 43 00D22 P.AAR: .ASCII \CDD\$\_ATTNOTFND\ ;

			0000	00000	.ENTRY	CDD_ATTNOTFND, Save nothing	2741
	5E		0C	C2	SUBL2	#12, SP	
04	AE		06	D0	MOVL	#6, CDDSHR	2742
08	AE	FC57	CF	9E	MOVAB	CDDSHR1, CDDSHR+4	
			0E	DD	PUSHL	#14	
04	AE	DE	AF	9E	MOVAB	P.AAR, NAMSTR+4	
		00000000'	EF	9F	PUSHAB	CDD_ATTNOTFND	2746
		04	AE	9F	PUSHAB	NAMSTR	
		10	AE	9F	PUSHAB	CDDSHR	
00000000G	00		03	FB	CALLS	#3, LIB\$FIND_IMAGE_SYMBOL	
	11		50	E8	BLBS	STATUS, 1\$	2747
			50	DD	PUSHL	STATUS	
			7E	D4	CLRL	-(SP)	
00000000G	00	001C11B4	8F	DD	PUSHL	#1839540	
			03	FB	CALLS	#3, SOR\$\$ERROR	
			04	00	RET		2748

; Routine Size: 62 bytes, Routine Base: SOR\$RO\_CODE + 0D30



CRE\_TRANS  
V04=000

User Literal Definitions

C 5  
16-Sep-1984 01:08:56  
14-Sep-1984 13:10:31

VAX-11 Bliss-32 V4.0-742  
[SORT32.SRC]CRETRANS.B32;1

Page 47  
(15)

: 1161 2749 1 END  
: 1162 2750 0 ELUDOM

PSECT SUMMARY

Name	Bytes	Attributes
SORSRO_CODE	2	NOVEC,NOWRT, RD, EXE, SHR, LCL, REL, CON, PIC,ALIGN(2)
SORSRW_PICDATA	44	NOVEC, WRT, RD, NOEXE,NOSHR, LCL, REL, CON, PIC,ALIGN(2)
SORSRO_CODE	3438	NOVEC,NOWRT, RD, EXE, SHR, LCL, REL, CON, PIC,ALIGN(2)

Library Statistics

File	----- Total	Symbols Loaded	----- Percent	Pages Mapped	Processing Time
\$255\$DUA28:[SYSLIB]STARLET.L32;1	9776	33	0	581	00:01.0
\$255\$DUA28:[SORT32.SRC]SORLIB.L32;1	409	123	30	34	00:00.3

: Information: 9  
: Warnings: 0  
: Errors: 0

COMMAND QUALIFIERS

: BLISS/CHECK=(FIELD,INITIAL,OPTIMIZE)/NOTRACE/LIS=LISS\$:CRETRANS/OBJ=OBJ\$:CRETRANS MSRC\$:CRETRANS/UPDATE=(ENH\$:CRETRANS)

: Size: 2812 code + 674 data bytes  
: Run Time: 00:53.2  
: Elapsed Time: 02:43.3  
: Lines/CPU Min: 3103  
: Lexemes/CPU-Min: 33389  
: Memory Used: 307 pages  
: Compilation Complete



0363

DIGITAL EQUIPMENT CORPORATION  
CONFIDENTIAL AND PROPRIETARY